

**Technická Univerzita Košice
Fakulta elektrotechniky a informatiky**

**Možnosti vizualizácie experimentálnych
údajov vo webovej rozšírenej realite**

Bakalárska práca

2022

Lukáš Mikula

**Technická Univerzita Košice
Fakulta elektrotechniky a informatiky**

**Možnosti vizualizácie experimentálnych
údajov vo webovej rozšírenej realite**

Bakalárska práca

Študijný program: Informatika
Študijný odbor: 9.2.1 Informatika
Školiace pracovisko: Katedra počítačov a informatiky (KPI)
Školiteľ: Ing. Štefan Korečko, PhD.
Konzultant:

Košice 2022

Lukáš Mikula

Abstrakt v SJ

Táto Bakalárska práca sa zameriava na riešenie problému používateľského rozhrania a zážitku v experimentálnom nástroji NDMVR. Jej cieľom je zlepšiť používateľský zážitok a rozšíriť možnosti interakcie s nástrojom ako pre desktopovú verziu nástroja, tak aj pre jeho prostredie vo virtuálnej realite. Na začiatku sa vykonala analýza možnosti používaných technológií pre rozšírenie používateľského rozhrania. Nadobudnuté poznatky sa následne odzrkadľujú v návrhu riešenia so zreteľom na praktickosť, rozširiteľnosť a používateľský zážitok. Po otestovaní, značnej úprave a implementovaní návrhu vzniká optimálne a ucelené riešenie. Vyhodnotenie sa zameriava na spracovanie výstupu z používateľského testovania a v závere práce sú načrtnuté možnosti rozšírenia a budúceho smerovania projektu.

Kľúčové slová v SJ

virtálna realita, rozšírená realita, A-Frame, React, UX, UI, používateľský zážitok, používateľské prostredie, histogram

Abstrakt v AJ

This bachelor thesis focuses on solving the problem of user interface and experience in the experimental tool NDMVR. Its goal is to improve the user experience and expand the possibilities of interaction with the tool both for the desktop version of the tool and for its environment in virtual reality. Initially, an analysis was made of the feasibility of the technologies used to extend the user interface. The acquired knowledge is then reflected in the design of the solution with regard to practicality, scalability and user experience. After testing, significant modification and implementation of the design, an optimal and comprehensive solution is created. The evaluation focuses on processing the output of user testing and at the end of the work outlines the possibilities of expansion and future direction of the project.

Kľúčové slová v AJ

virtual reality, extended reality, A-Frame, React, UX, UI, user experience, user interface, histogram

Bibliografická citácia

MIKULA, Lukáš. *Možnosti vizualizácie experimentálnych údajov vo webovej rozšírenej realite*. Košice: Technická Univerzita Košice, Fakulta elektrotechniky a informatiky, 2022. 40s. Vedúci práce: Ing. Štefan Korečko, PhD.

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra počítačov a informatiky

ZADANIE
BAKALÁRSKEJ PRÁCE

Študijný odbor: **Informatika**

Študijný program: **Informatika**

Názov práce:

**Možnosti vizualizácie experimentálnych údajov vo webovej
rozšírenej realite**

**Possibilities of Experimental Data Visualization in Web-based Extended
Reality**

Študent: **Lukáš Mikula**

Školiteľ: **Ing. Štefan Korečko, PhD.**

Školiace pracovisko: **Katedra počítačov a informatiky**

Konzultant práce:

Pracovisko konzultanta:

Pokyny na vypracovanie bakalárskej práce:

1. Oboznámiť sa so súčasným stavom softvérového komponentu NDMVR pre vizualizáciu experimentálnych údajov v rozšírenej realite.
2. Analyzovať aktuálne prístupy k vizualizáciám experimentálnych údajov v rozšírenej realite.
3. Identifikovať nedostatky komponentu NDMVR z hľadiska používateľského zážitku a to najmä so zreteľom na použitie virtuálno-reálnych prilieb.
4. Navrhnuť riešenie vybraných nedostatkov a implementovať ich do komponentu NDMVR.
5. Implementované riešenie overiť z hľadiska výkonu a používateľského zážitku.
6. Vypracovať dokumentáciu podľa pokynov vedúceho práce.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 27.05.2022

Dátum zadania bakalárskej práce: 29.10.2021



A handwritten signature in blue ink, appearing to be "Liberios Vokorokos".

prof. Ing. Liberios Vokorokos, PhD.

dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som záverečnú prácu vypracoval(a) samostatne s použitím uvedenej odbornej literatúry.

Košice, 13.5.2022

.....

Vlastnoručný podpis

Podakovanie

Na tomto mieste by som rád poďakoval svojmu vedúcemu práce za jeho čas a odborné vedenie počas riešenia mojej záverečnej práce, doménovému expertovi RnDr. Martinovi Vaľovi, PhD. za cenné poznatky a skúsenosti, bez ktorých by bolo veľmi ťažké dosiahnuť tieto výsledky a Bc. Martinovi Feketemu za projekt NDMVR a pomoc pri implementácii riešenia.

Rovnako by som sa rád poďakoval svojim rodičom a priateľke za ich podporu a povzbudzovanie počas celého môjho štúdia.

V neposlednom rade by som sa rád poďakoval pánom *Donaldovi E. Knuthovi* a *Leslie Lamportovi* za typografický systém \LaTeX , ktorý mi signifikantne pomohol pri písaní práce.

Obsah

Úvod	1
1 Analytická časť	3
1.1 Analýza aktuálneho stavu projektu NDM VR	3
1.1.1 Analýza podstránky THProjection	3
1.1.2 Analýza podstránky TH3React	7
1.1.3 Zhodnotenie analýzy aktuálneho stavu projektu NDMVR	8
1.2 Analýza používaných technológií	8
1.2.1 Virtuálna realita	8
1.2.2 Softvérový rámec A-Frame	9
1.2.3 Vstupno-výstupné zariadenia a úroveň interaktivity z pohľadu využitia dostupných technológií	9
1.2.4 Otestovanie použiteľnosti NDMVR na iných zariadeniach	10
1.3 Analýza aktuálnych prístupov k vizualizácií experimentálnych údajov v rozšírenej realite	12
1.3.1 Project NEO	12
1.3.2 Preloženie veľkých dát do VR vizualizácie	13
1.3.3 Vyhodnotenie Analýzy aktuálnych prístupov k vizualizácií experimentálnych údajov v rozšírenej realite	13
1.4 Analýza súčasného stavu VR aplikácií	13
1.4.1 Gravity Sketch	14
1.4.2 Brushwork VR	16
1.4.3 Zhodnotenie analýzy súčasného stavu VR aplikácií	17
1.5 Analýza dostupných technológií	17
1.5.1 A-Frame htmlembded Component	17
1.5.2 Zhodnotenie analýzy dostupných technológií	19
2 Syntetická časť	20
2.1 Návrh implementácie Tabletů	20

2.1.1	Princíp používania tabletu	20
2.1.2	Problém s interakciou	21
2.1.3	Tablet pre VR zariadenia	22
2.1.4	Funkcionalita a nástroje dostupné prostredníctvom tabletu .	23
2.1.5	Návrh implementácie nástrojov	24
2.1.6	Vizuálny návrh tabletu	25
2.2	Implementácia tabletu	26
2.2.1	Komponent DesktopTablet()	26
2.2.2	Komunikácia tabletu s inými komponentami	31
2.2.3	Zefektívňovanie komunikácie s tabletom	32
2.2.4	Zmena v implementácii nástrojov	32
2.2.5	Funkcia renderOculusTablet()	33
2.2.6	Implementácia tabletu pre desktop	33
2.3	Pridávanie nástroja	33
2.3.1	Pridávanie nového nástroja	34
3	Vyhodnotenie	36
3.1	Používateľské testovanie	36
3.1.1	Vyhodnotenie SUS metriky	37
3.1.2	Vyhodnotenie NASA TLX metriky	37
3.1.3	Vyhodnotenie používateľského testovania	37
4	Záver	38
	Literatúra	39
	Zoznam skratiek	41
	Zoznam príloh	42

Zoznam obrázkov

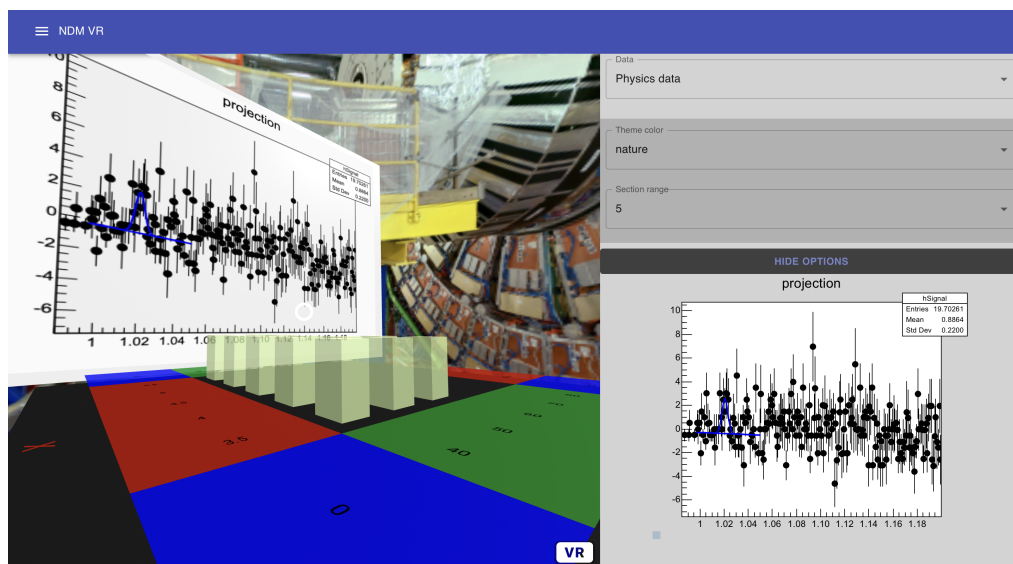
1	Screenshot z programu NDM VR	1
1.1	Výsek z webovej aplikácie NDM VR s otvoreným menu	3
1.2	Projekčný mód v programe TH3react	5
1.3	Pomocník v programe TH3react	6
1.4	Screenshot z pod-stránky TH3react	8
1.5	VR Box ZEISS VR ONE (ZDROJ: [9])	10
1.6	Screenshot z projektu NDMVR otvoreného na mobilnom zariadení vo VR móde.	11
1.7	Screenshot z náhodného VR kompatibilného videa z portálu Youtube. (ZDROJ: [10])	11
1.8	Screenshot zo softvéru Project NEO z procesu definovania kvádrovej selekcie.	12
1.9	Screenshot z programu Gravity Sketch (ZDROJ: [12])	14
1.10	Používateľské rozhranie aplikácie Gravity Sketch pre dvojrozmerné elementy	15
1.11	Gesto na zmenu mierky zobrazenia scény v programe Gravity Sketch	15
1.12	Hlavné menu v aplikácii Brushwork VR	17
2.1	Prvý prototyp tabletu pre desktopovú verziu webu. (Dizajn nie je finálny)	21
2.2	Diagram prípadov použitia tabletu.	22
2.3	Dizajn NDMVR tabletu.	25
2.4	Stromová štruktúra adresára src projektu NDMVR.	27
2.5	Obsah komponentu DesktopTablet.	28
2.6	Komponent diagram DesktopTabletu a relevantných komponentov.	29
2.7	Zjednodušená verzia desktopového tabletu pre VR zariadenia.	31

Úvod

Urýchľovač častíc dokáže pri rôznych fyzikálnych experimentoch vygenerovať veľké množstvo komplexných dát. Fyzici následne pracujú s týmito dátami a na základe ich spracovania vyvodzujú závery a objavy.

Dáta sú prezentované vo forme histogramov. Zobrazovacími objektami histogramov sú takzvané biny. Podľa Karla Pearsna [1] biny reprezentujú sériu intervalov vytvorenú z množiny hodnôt, ktoré sa neprekrývajú a môžu ale nemusia byť rovnakej veľkosti. Pri intervaloch rovnakej veľkosti, sa nad intervalom zobrazuje obdĺžnik, ktorého výška zodpovedá frekvencii výskytu daných hodnôt v konkrétnom intervale.

Podľa slov RNDr. Martina Vaľu, PhD. táto práca si vyžaduje sledovanie dát v rôznych mierkach, ich rozdeľovanie, vnímanie aj malých odchýlok v hodnotách binov. Zjednodušeniu tohto procesu sa venuje Bc. Martin Fekete vo svojej práci Vizualizácia experimentálnych údajov v zdieľanej rozšírenej realite a jej používateľské rozhranie[2].



Obr. 1: Screenshot z programu NDM VR

Dôležitým faktorom je interakcia používateľa s dátami. Kedysi práca s dátami bola možná len prostredníctvom klasických vstupných zariadení, ako je myš,

klávesnica, po prípade nejakej špeciálnej, z pravidla na trhu ťažko dostupné ovládače. Dnes nám však technológie umožňujú omnoho viac. Za použitia okuliarov pre virtuálnu realitu a k tomu prislúchajúcich ovládačov sa nám otvárajú možnosti práce s dátami vo virtuálnej realite. Za pomoci práve tejto technológie je používateľ schopný ovládať, rotovať, škálovať objekty vo virtuálnej realite, dokonca sa môže okolo nich premiestňovať a pozorovať ich z rôznych uhlov. Virtuálna realita nám ponúka na jednej strane naozaj vysokú úroveň interakcie s virtuálnymi objektami, a tak isto aj pohlcujúci zážitok pre používateľa. Pojmy ako virtuálna realita, augmentovaná realita a zmiešaná realita sa podľa článku [3] v dnešnej dobe skloňujú stále viac, čo značí rastúcemu trendu softvérových produktov práve pre tieto platformy.

Formulácia úlohy

Projekt Bc. Martina Feketeho [2] je softvérový nástroj existujúci v trojrozmernom priestore, teda sa dá zadefinovať ako hybrid medzi klasickým spravidla dvojrozmerným počítačovým programom, a kanceláriou resp. pracovným prostredím v reálnom svete. Náplňou tejto práce je projekt zefektívniť a zadefinovať prístup k riešeniu používateľského rozhrania v trojdimenzionálnom priestore.

Výsledkom tejto práce je front-end webovej aplikácie. Keďže je to projekt už existujúci, a my ho budeme len modifikovať, technológie máme vopred zadefinované. Budeme používať web framework A-Frame¹, ReactJS² a JS ROOT[4]. Primárny hardware, pre ktorý bude táto práca konštruovaná je OCULUS Quest 2³.

Z hľadiska interakcie, je potrebné zistiť všetky doposiaľ človekom zadefinované spôsoby interakcie so softvérom v realite.

Z vizuálneho hľadiska, keďže projekt je spomínaný hybrid dvoj a troj dimenzióneho interface-u, potrebujeme zistiť ako správne navrhnuť jedno ale aj druhé, aby sme dokázali z dizajnovacích zvyklostí oboch platforiem využiť silné a prínosné stránky a teda navrhnuť a implementovať vlastné riešenie.

Prvým krokom bude analýza projektu NDMVR do hĺbky, potom analyzujeme možnosti technológií ktoré v projekte máme použiť a tak isto analyzujeme aktuálny stav. Medzi popredné problémy, ktoré je potrebné v tomto projekte vyriešiť patrí prenos dvojrozmerného segmentu projektu do VR, navrhnutie nástrojov pre lepšiu manipuláciu dát a spraviť celkové ovládanie intuitívnejším.

¹<https://www.aframe.io/>

²<https://www.reactjs.org/>

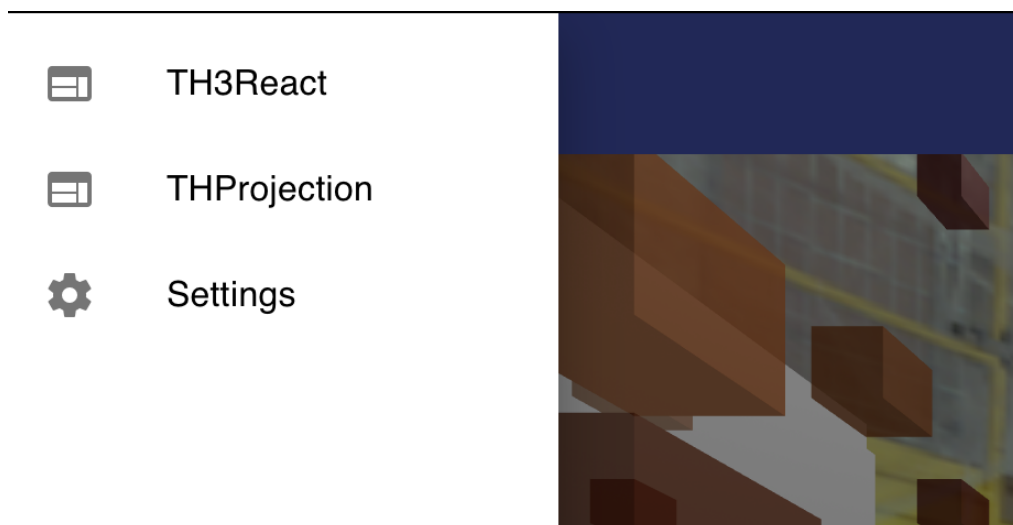
³<https://www.oculus.com/quest-2/>

1 Analytická časť

V prvom rade potrebujeme analyzovať projekt NDM VR, zistiť ako sa vlastne ovláda, a aké možnosti sú od tohto používateľského rozhrania očakávané. Následne prevedieme rozbor použitých technológií.

1.1 Analýza aktuálneho stavu projektu NDM VR

Ako môžeme vidieť na obrázku 1, používateľské rozhranie sa skladá z dvoch rozdielnych okien. Tými sú VR zobrazenie, a JS ROOT bočný panel. Okrem týchto dvoch markantných elementov sa tu nachádza aj menu 1.1, ktoré po rozkliknutí ponúka prepnutie sa medzi projektami TH3react, THProjection a nastaveniami.



Obr. 1.1: Výsek z webovej aplikácie NDM VR s otvoreným menu

1.1.1 Analýza podstránky THProjection

Sekcia THProjection 1 nám poskytuje ako okno v virtuálnej realite, tak aj panel na pravej strane s možnosťami na výber zdroju dát, ktoré si používateľ chce zobraziť, farebnej schémy, ktorú si želá použiť a rozsah v ktorom sa mu histogram

zobrazí. V spodnej časti sa nachádza zobrazenie histogramu v 2D, a niekoľko nástrojov na prispôsobenie a ovládanie tohto zobrazenia. Možnosti používateľa v trojrozmernom priestore projektu sú:

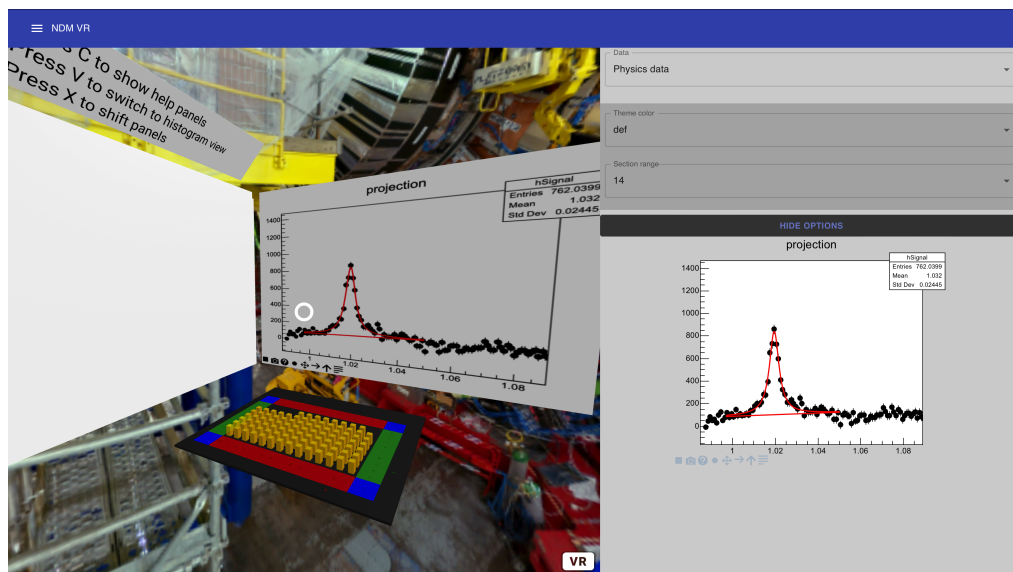
- Pohyb po ploche s klávesami W, S, A, D, alebo so šípkami
- Neobmedzená rotácia hlavy do všetkých smerov s myškou, alebo v prípade VR headsetu s rotáciou hlavy
- Prepínanie sa medzi deviatimi vopred nastavenými pozorovacími pozíciami a uhlami s klávesami 1 až 9
- Pohyb po ose Y za pomoci kláves Q a E
- Zmena zobrazeného pásma na histograme, pohyb naprieč všetkými osami v oboch smeroch za použitia kláves U, I, O, J, K, L
- Zvolenie si konkrétneho binu s ľavým tlačidlom na myši.
- Prechod do projekčného módu, alebo pomocníka. Zapnutie a vypnutie sa realizuje s klávesou V
- Prechod v rámci projekčného módu medzi projekciami a pomocníkom pomocou klávesy C
- Otáčanie plátien v projekčnom móde s klávesou X

Projekt je stále vo vývoji, teda nie všetky funkcie sú plne k dispozícii. Avšak pre nás je práve dnešný stav projektu východisková pozícia pre túto bakalársku prácu.

Pohyb používateľa vo virtuálnom prostredí je vcelku intuitívny, ovládanie pohybu je mapované v súlade so štandardmi a konvenciami HCI[5]. V prípade zmeny renderovaného úseku histogramu nám zabralo niekoľko pokusov aby sme zistili, že na klávesách I, K, J, L je vlastne mapovaný štandardný šípkový kríž. V pomocníkovi to nie je prezentované ako šípkový kríž a používateľ sa k tomuto faktoru sám nedopracuje. Pri presune renderovaného výseku histogramu je zreteľne viditeľné, že sa scéna pohla. No používateľovi nie je úplne jasne ukázané čo sa zmenilo. Po jednom stlačení klávesy do ľubovoľného smeru presunu zobrazeného výseku pribudne v histograme práve jeden nový rad hodnôt daným smerom, a z opačného konca jeden rad ubudne, zatiaľ čo merítko na osi sa priamoúmerne prispôsobí. Mapovanie vykonávania práve tejto akcie nie je dobre vyriešené, pretože pokiaľ si používateľ nepamätá presný interval ktorý mal pred ak-

ciou v danej osi zobrazený, nie je možné spoznať rozdiel, a teda v prípade náhodného stlačenia jednej z kláves určených na práve tento pohyb to používateľ nevie vrátiť späť, ani zistiť že čo sa zmenilo keďže scéna vyzerá rovnako ako predtým. Sú prípady kedy rozdiely v jednotlivých binoch sú dostatočne veľké aby si používateľ dokázal všimnúť zmenu na samotnom histograme, no dáta môžu byť rôznych hodnôt, z toho vyplývajú biny rôznych rozmerov a teda rozdiel nemusí byť stále markantný. Z týchto dôvodov by bolo lepšie ovládať presun renderovaného intervalu za pomoci tlačidiel vo virtuálnom prostredí, nie klávesových skratiek. Používateľ si bude vedomý čo robí, v ktorom smere mení zobrazenie a keďže by bolo tlačidlo viditeľné a dostatočne seba-vysvetľujúce, predišlo by sa tým aj náhodnému stlačeniu.

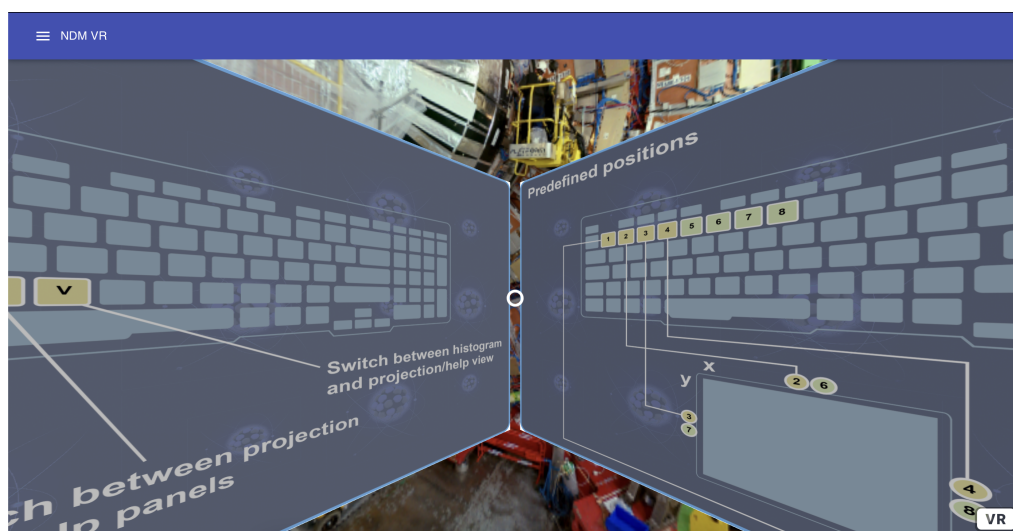
Interval renderovaného histogramu začína stále na pozícii 0, 0, 0. Pri zmene zobrazovaného intervalu sa celá scéna posunie daným smerom, až na používateľa. Čo znamená že zmena zobrazovaného intervalu si vyžaduje aj priebežné doladenie lokácie používateľa, keďže bez priameho pohľadu na merítka osí nemá odkiaľ vedieť na ktorý výsek dáť sa pozerá, čo značne komplikuje prácu s nástrojom. Okrem toho os Z nemá mierku vôbec, takže používateľ v jednom smere nevie na akej úrovni sa nachádza.



Obr. 1.2: Projekčný mód v programe TH3react

To, čo sme v záujme zreteľnosti a čitateľnosti tejto práce nazvali projekčný mód, takto nazvané ani prezentované na stránke nie je, ale správa sa to tak. Klávesy X a C vo virtuálnom prostredí nevykonávajú žiadnu akciu. Až po stlačení klávesy V, kedy sa používateľ presunie do zobrazenia projekcií^{1.3}, má možnosť si s ich pomocou otáčať projekcie a prepínať sa medzi pomocníkom a projekciami.

Niekedy sa používateľovi po stlačení klávesy V zobrazia projekcie, inokedy pomocník, čo môže priviesť používateľa na slepú cestu, lebo ak sa mu pri prvom kontakte s projektom po stlačení klávesy V zobrazia panely s projekciami, nemá odkiaľ vedieť že za pomoci klávesy C sa vie prepnúť do pomocníka s vysvetlením jednotlivých kláves. Správanie sa týchto ovládacích prvkov je totižto navrhnuté tak, že funkčnosť kláves C a X nekoreluje s faktom, či projekčný mód je zapnutý, alebo vypnutý. Rozdiel je len v tom, že v projekčnom móde používateľ zmeny vidí, v bežnej simulácii nie. Čo môže jednoducho dostať používateľa do scenára, v ktorom nedostáva žiadne informácie o vykonanej akcii. V projekčnom móde 1.2 sa používateľovi ukázu dve prázdne biele plátna. Je to z toho dôvodu, že používateľ si zatiaľ nezvolil konkrétny bin v histograme, ktorý si želá zobrazíť.



Obr. 1.3: Pomocník v programe TH3react

Z práce s programom a po konzultáciách s jeho autorom sme zistili, že program je v tomto štádiu navrhnutý pre nasledovný scenár použitia:

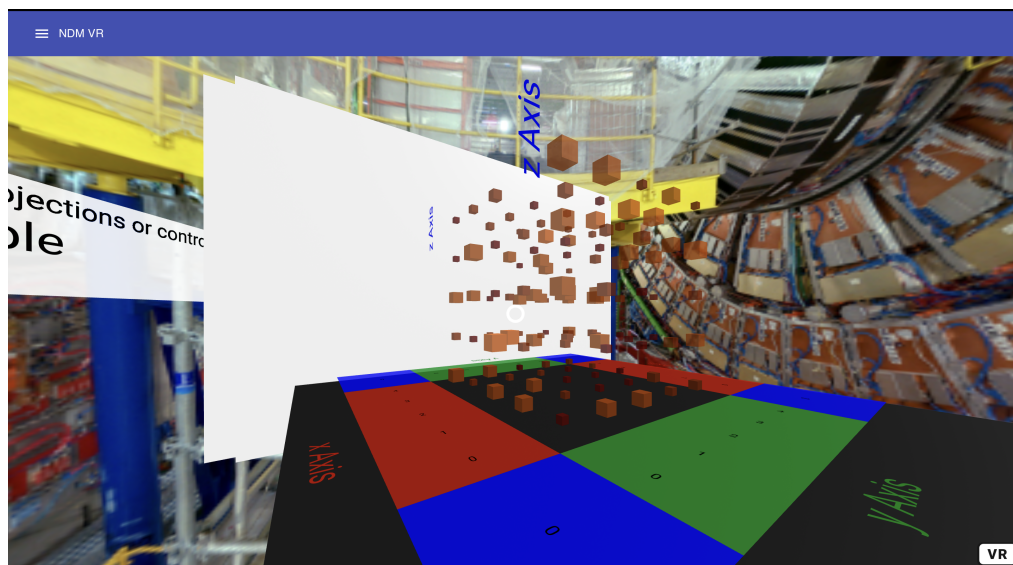
1. Používateľ si otvorí THProjection
2. V paneli napravo si zvolí zdroj zobrazovaných dát, po prípade farebnú schému a rozsah zobrazovaného výseku dát
3. Program zobrazí histogram zvolených dát, nastavený na pozíciu 0, 0, 0
 - V tomto bode scenáru si používateľ môže zmeniť zobrazovaný výsek dát pomocou kláves I, J, K, L
4. Používateľ si zvolí požadovaný bin
5. program mu zobrazí 2D graf hodnôt obsiahnutých v danom bine na:

- Tabuli za histogramom vo VR
 - V 2D elemente stránky, teda v paneli napravo
 - v projekčnom móde
6. Používateľ môže v zobrazenom grafe v 2D elemente stránky za pomoci nástrojov z knižnice JSROOT:
- Pohybovať sa v zobrazenom grafe
 - Zobrazíť si informácie k jednotlivým hodnotám
 - Zmeniť si spôsob zobrazenia
 - zobrazíť graf na celú obrazovku
 - priblížiť si graf na žiadanú sekciu
 - vyhotoviť snímku grafu

Napriec všetkými bodmi scenára ktoré zahŕňajú interakciu so sekciou stránky obsahujúcou virtuálnu realitu sa používateľ vo virtuálnom prostredí môže voľne pohybovať a rotovať. Nedostatkom tohto riešenia je zobrazenie zvoleného binu vo virtuálnom prostredí. Prostredie nijako neoznačuje bin, s ktorým sa pracuje, a používateľ veľmi ľahko o tom stratí prehľad. Tak isto pri manipulácii s 2D zobrazením obsahu binu za pomoci nástrojov z knižnice JSROOT [4] sa vykonané zmeny nezobrazujú na plátne za histogramom vo virtuálnom prostredí, ani v projekčnom móde.

1.1.2 Analýza podstránky TH3React

Kľúčovou vlastnosťou podstránky TH3React 1.4 je, že zobrazuje TH3 histogram a nie TH2. Sekundárne slúži hlavne na zobrazenie všetkých možností, aké projekt ponúka. Pointou nie je byť praktickým použiteľným nástrojom, je to skôr experimentálne zobrazenie doposiaľ dosiahnutých funkcionalít v projekte. Na obrázku 1.4 môžeme vidieť zobrazenie 5 rozmerného histogramu vo virtuálnom prostredí. Na osi Z je naruozdiel od THProjection obsiahnutá ďalšia dimenzia. Biny sú kocového tvaru, a ako ich farba, tak isto aj ich veľkosť zobrazujú samostatný rozmer histogramu. Dáta sa v čase menia, čo je viditeľné na meniacej sa veľkosti a farbe jednotlivých binov. Stránka obsahuje plnú funkcionalitu simulačného panela z THProjection.



Obr. 1.4: Screenshot z pod-stránky TH3react

1.1.3 Zhodnotenie analýzy aktuálneho stavu projektu NDMVR

Na základe analýzy projektu sme objavili niekoľko nedostatkov v UX a UI, tak isto chýbajúce implementácie niektorých funkcionalít vo verzii pre zariadenia Oculus. Po konzultácii s vedúcim práce sme zistili, že pre projekt bude oveľa prínosnejšie sa zamerať práve na problém chýbajúcej funkcionality vo verzii pre zariadenia Oculus.

1.2 Analýza používaných technológií

Táto časť analýzy slúži na rozbor technológií použitých v projekte NDMVR. Keďže táto práca priamo nadväzuje na spomínaný projekt, technológie, prostredníctvom ktorých realizujeme riešenie máme vopred zadané.

1.2.1 Virtuálna realita

Virtuálna realita VR je podľa zdroja [3] softvérová simulácia trojrozmerného priestoru. Na interakciu v nej používame buď klasickú zostavu, obrazovku, myš a klávesnicu, alebo špeciálne stereoskopické zariadenia spolu s ich ovládačmi, ktoré dokážu používateľovi simulovať oveľa väčší zážitok. Dnes sa vďaka svojim zahlbujúcim prednostiam využíva hlavne v hernom priemysle. Spolu s týmto termínom sa spájajú aj pojmy rozšírená realita AR a zmiešaná realita MR. Podľa článku[6] rozšírená realita (AR z anglického augmented reality) je technológia, ktorá priamo alebo nepriamo zobrazuje fyzický reálny svet ktorý bol rozšírený

o digitálne dáta. Objekt vložený do reálneho sveta je potom možné sledovať prostredníctvom obrazovky zariadenia. Túto technológiu v dnešnej dobe využíva väčšina smartfónov.

Zmiešaná realita (MR z anglického mixed reality) označuje spojenie fyzickej reality a VR kde obe reality na seba navzájom reagujú v reálnom čase. V tomto prostredí je možné vytvárať kontakt ako s fyzickými objektami na scéne, tak aj s virtuálnymi. Na interakciu nie sú potrebné žiadne špeciálne ovládače, používateľ môže použiť ruky alebo nohy. MR VR a AR zariadenia majú väčšinou na prednej strane zakomponované kamery a senzory, vďaka ktorým snímajú okolie a prípadný pohyb.

1.2.2 Softvérový rámec A-Frame

V tejto sekcii sa budeme venovať analýze možností softvérového rámca A-Frame. Podľa dokumentácie[7] A-Frame je webový softvérový rámec pre vytváranie VR zážitkov. A-Frame rozširuje HTML, čo umožňuje jednoduchú prácu s týmto softvérovým rámcom. Aj napriek podobnosti s HTML, A-Frame nie je len značovací jazyk, ale obsahuje jadro implementujúce softvérový architektonický vzor na Entity-Component-System, ktorý poskytuje deklaratívnu a rozšíriteľnú štruktúru knižnice Three.js [8]. Samotný A-Frame má jednu z najväčších VR komúnít a obsahuje rozšírenia a podporu pre VR prilby ako Vive, Rift, Windows Mixed Reality, Oculus go, Daydream, GearVR a rovnako aj pre rozšírenú realitu AR.

1.2.3 Vstupno-výstupné zariadenia a úroveň interaktivity z pohľadu využitia dostupných technológií

Ako je spomínané v úvode tejto práce, projekt je konštruovaný primárne pre zariadenie Oculus Quest 2. Je to hlavne z dôvodu širokej škály interakčných možností ktoré ponúka za relatívne nízku cenu. Oculus Quest 2 ponúka 6 stupňov voľnosti[3], prvé 3 stupne popisujú možnosť rotácie hlavy v osiach x, y, z. Druhé 3 stupne reprezentujú možnosť pohybu do všetkých smerov. Okrem toho zariadenie sníma aj polohu, rotácie a stav tlačidiel na jeho ovládačoch. Všetky tieto vlastnosti nám otvárajú naozaj široké spektrum možností na preloženie pre človeka intuitívnych interakcií do virtuálnej reality.

1.2.4 Otestovanie použiteľnosti NDMVR na iných zariadeniach

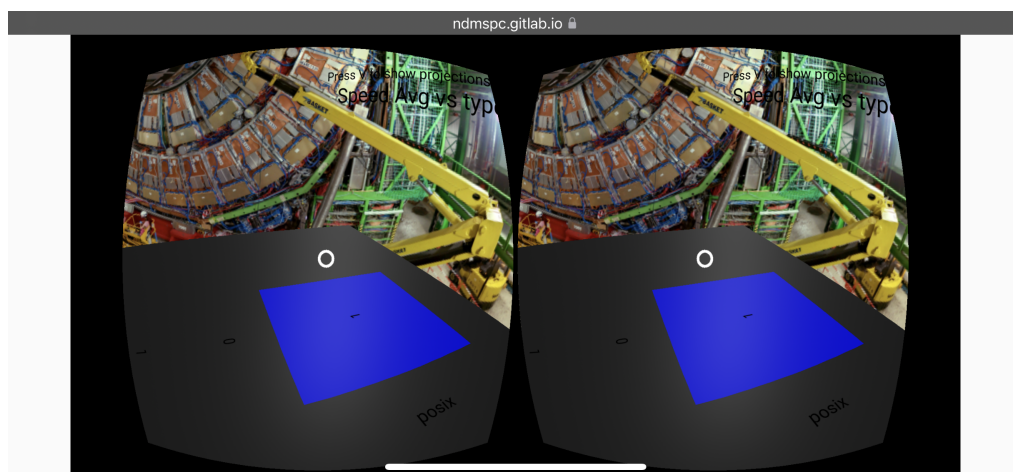
Aj keď sú dnes VR zariadenia cenovo oveľa dostupnejšie ako kedysi, stále nie sú pre širokú verejnosť tak dostupné, ako VR Boxy. Princíp VR boxov spočíva v simulácii zážitku na úrovni VR za použitia inteligentného telefónu. VR box je sám o sebe spravidla pasívne zariadenie, ktoré obsahuje optiku, a slot dedikovaný pre inteligentné zariadenie. Za použitia VR kompatibilnej aplikácie a zariadenia, ktoré je kompatibilné s VR boxom, zážitok vie byť porovnateľný. Najsilnejšou stránkou VR boxov je ich cena. Zatiaľ čo VR headsety sa hýbu v stovkách eur, ceny väčšiny VR boxov sú do 100 eur. Najväčšou nevýhodou VR boxov je ich nedostatok interaktivity. Keďže sú to pasívne zariadenia, neobsahujú ovládače a teda pre dosiahnutie interakcie si musí ovládač používateľ zakúpiť samostatne. Avšak niektoré VR boxy obsahujú jedno tlačidlo, ktoré po stlačení vykoná dotyk na jednej fixnej pozícii na obrazovke vloženého zariadenia. Ale keďže NDMVR podporuje ako vstup aj myš a klávesnicu, rozhodli sme sa nástroj otestovať v hardvérovej konfigurácii smartphone, bluetooth myš, bluetooth klávesnica a VR Box. V teoretickej rovine by sme mali dosiahnuť rovnakú konfiguráciu ako v prípade interakcie za použitia stolného počítača, ale vo VR. Pre tento test sme sa rozhodli použiť zariadenie Zeiss VR ONE spolu s Apple iPhone 13 mini.



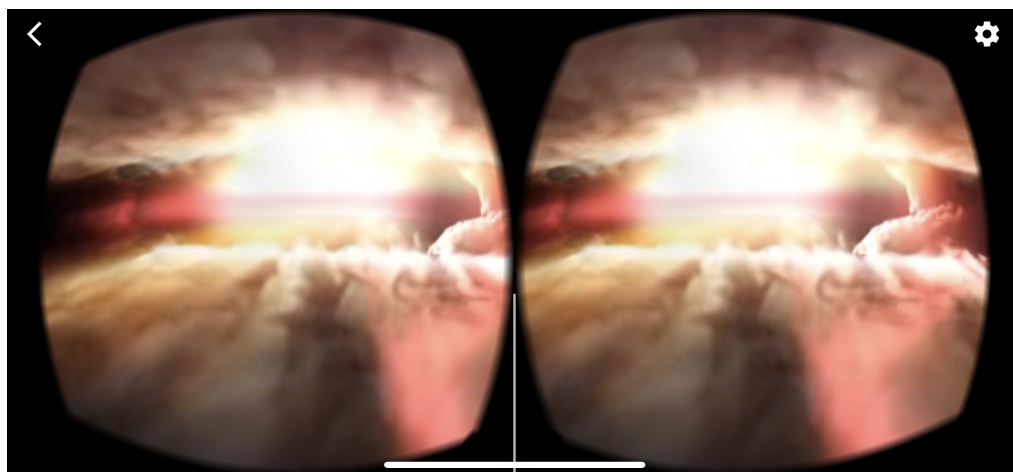
Obr. 1.5: VR Box ZEISS VR ONE (ZDROJ: [9])

Po otvorení stránky NDMVR na mobilnom zariadení v režime VR a po nasadení VR boxu, bolo viditeľné silné rozmazanie. Jednotlivé segmenty displeja, každý určený pre jedno oko, boli nezarovnané so šošovkami, čo vytváralo silne nepríjemný zážitok. Obsah stránky nebol viditeľný, nedalo sa naň ani dobre zaostriť. Po páru sekundách bol z test ukončený. Z dôvodu nezrovnaností premie-

taných plôch so šošovkami bol projekt nepoužiteľný.



Obr. 1.6: Screenshot z projektu NDMVR otvoreného na mobilnom zariadení vo VR móde.



Obr. 1.7: Screenshot z náhodného VR kompatibilného videa z portálu Youtube. (ZDROJ: [10])

Keď sa pozrieme na porovnanie snímok obrazovky zo zariadenia zo stránky projektu a z náhodného VR kompatibilného videa z portálu Youtube, vidíme že tvar jednotlivých zobrazovacích plôch je rozličný. Na snímku obrazovky z projektu (Obr. 1.6) môžeme vidieť značne väčší rám po bokoch zobrazovanej plochy. Stránka sa nezobrazila na celú šírku zariadenia tak, ako tomu je v prípade videa (Obr. 1.7), čo môže mať za následok zdeformované premietacie plochy, a teda nepoužiteľnosť projektu prostredníctvom VR boxu.

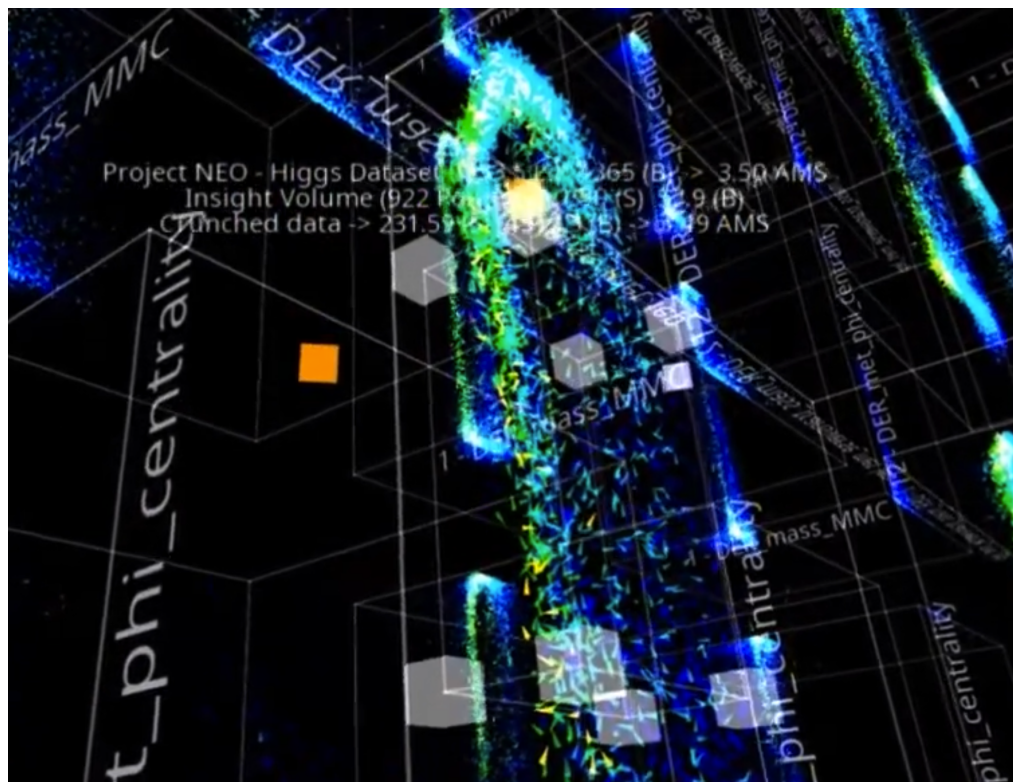
Riešením môže byť upravenie stránky projektu tak, aby pri otočení mobilného zariadenia na šírku dokázala využiť celú plochu zariadenia, ktorú má k dispozícii.

1.3 Analýza aktuálnych prístupov k vizualizácií experimentálnych údajov v rozšírenej realite

V tejto časti analýzy sa pozrieme na existujúce riešenia zobrazenia experimentálnych údajov. Náš primárny zámer je preskúmať interakčné možnosti nástrojov.

1.3.1 Project NEO

Project NEO [11] je nástroj slúžiaci na zobrazenie dát pre machine learning vo VR. Francois Bertrand demonštruje aplikáciu, v ktorej sa súbor dát zobrazuje súčasne vo viacerých korelačných diagramoch a používateľ ich môže preskúmať vo VR. Softvér umožňuje prepínanie medzi trojdimenzionálnou reprezentáciou a šesťdimenzionálnou reprezentáciou, ktorá je realizovaná prostredníctvom tesseractu. V danom štádiu projekt podporoval zobrazenie dát z vlastného zdroja, pohyb medzi dátami, zoom a tak isto vytvorenie kvádrovej selekcie dát a následné vypísanie informácií o danej selekcii (Obr. 1.8). Detailnejšie informácie o podporovaných funkcionalitách, návrhu riešenia, alebo použitých technológiách nie sú zverejnené.



Obr. 1.8: Screenshot zo softvéru Project NEO z procesu definovania kvádrovej selekcie.

1.3.2 Preloženie veľkých dát do VR vizualizácie

Študenti z Univezrity Dayton vytvorili projekt¹, ktorý slúži na zjednodušenie získavania a spracovania geotechnických dát. Cieľom projektu je zautomatizovať získavanie týchto dát, ich transformácia pomocou machine learningu do podoby zrozumiteľnej pre človeka a následná vizualizácia týchto dát do VR a následné vymodelovanie tohto terénu vo VR na základe získaných informácií. Detailnejšie informácie o podporovaných funkcionalitách, návrhu riešenia, alebo použitých technológiách nie sú zverejnené.

1.3.3 Vyhodnotenie Analýzy aktuálnych prístupov k vizualizáciám experimentálnych údajov v rozšírenej realite

V tejto časti analýzy sme zistili, že zatiaľ existuje veľmi malé množstvo VR aplikácií určených na zobrazenie a prácu s experimentálnymi dátami. Podarilo sa nám nájsť dva projekty, no ani jeden z nich neponúka dostatočné množstvo informácií o projekte, aké by bolo potrebné pre detailnejšiu analýzu a teda nadobudnutie poznatkov.

1.4 Analýza súčasného stavu VR aplikácií

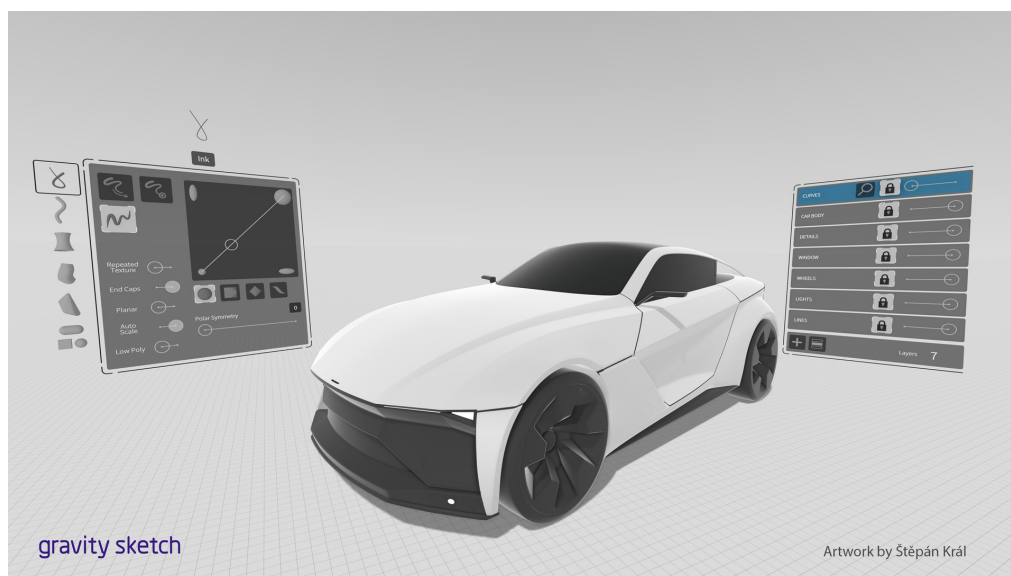
Pri práci s aplikáciami nám príde vcelku normálne ich ovládať prostredníctvom dvojrozmerného používateľského rozhrania. Elementy ako menu, záložky, nástroje, nastavenia, vyhľadávanie, majú svoj štandardizovaný tvar, vieme ako ich rozpoznať a narábať s nimi. V prípade trojrozmerných programov je situácia úplne iná, čo prináša viacero ako efektívnych, tak aj menej efektívnych možností pre vizualizáciu ovládacích elementov. Každopádne, prebratie zaužívaných štandardov dizajnu grafického používateľského prostredia neposkytuje riešenie pre všetky stavy do ktorých sa trojrozmerný program môže dostať. Preto sa v rozbere relevantných softvérových produktov sa budeme zameriavať na nasledujúce kritériá:

- Riešenie 2D elementov ako napríklad menu, nastavenia
- Spôsob interakcie s 2D elementami
- Panel s nástrojmi
- Pohyb po scéne

¹<https://www.youtube.com/watch?v=9fIyHN6an7k>

1.4.1 Gravity Sketch

Gravity Sketch² je nástroj slúžiaci na kreslenie a navrhovanie dizajnov vo VR. Ponúka širokú škálu nástrojov ako na priestorové kreslenie, tak aj na vytváranie preddefinovaných geometrických útvarov.



Obr. 1.9: Screenshot z programu Gravity Sketch (ZDROJ: [12])

V programe Gravity Sketch sú všetky dvojrozmerné elementy riešené spôsobom imaginárneho tabletu, ktorý sa používateľovi zobrazí pri ľavom ovládači stlačením klávesy X, a zmizne pri jej opätovnom stlačení. Používateľ pracuje s tabletom prostredníctvom pravého ovládača. Na ľavej strane tabletu sú znázornené jednotlivé nástroje formou trojrozmerného objektu, ktorým používateľ s daným nástrojom vytvorí a po rozkliknutí sa mu otvorí karta s rozšírenými nastaveniami daného nástroja.

Rovnakým spôsobom sú riešené aj nastavenia aplikácie, s jedným rozdielom, daný tablet je zavolaný stlačením iného tlačidla.

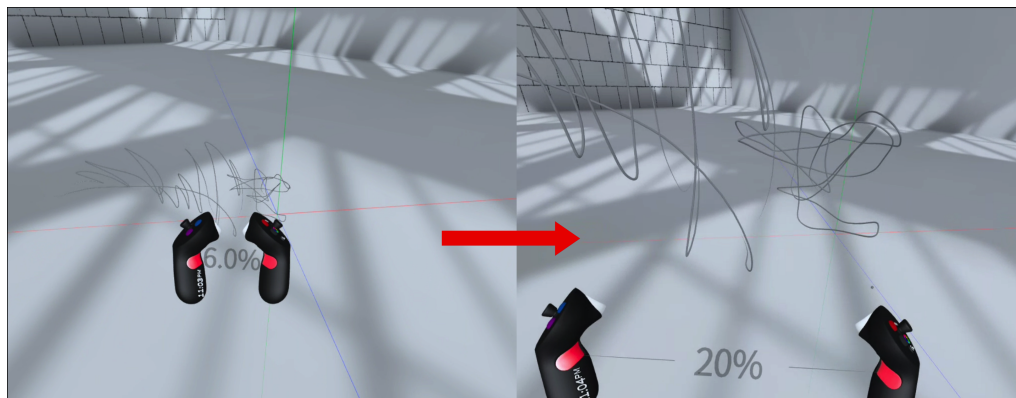
Medzi ostatné zaujímavé funkcie tohto programu patrí aj riešenie pohybu v priestore. Namiesto toho, aby sa používateľ musel presúvať po scéne, po stlačení tlačidla pod prostredníkom na ľavej ruke dokáže rotovať a presúvať celú scénu okolo tohto ovládača, teda okolo svojej ľavej ruky. Ak k tomu stlačí rovnaké tlačidlo na pravom ovládači, otvorí sa mu možnosť na zmenu mierky zobrazenia celej scény (Obr. 1.11), ktorá sa v tom momente mení približovaním a vzdďalovaním ovládačov.

Medzi zaujímavé prvky dizajnu používateľského prostredia v Gravity Sketch

²<https://www.gravitysketch.com>



Obr. 1.10: Používateľské rozhranie aplikácie Gravity Sketch pre dvojrozmerné elementy



Obr. 1.11: Gesto na zmenu mierky zobrazenia scény v programe Gravity Sketch

patrí jednoznačne hĺbka všetkých ikoniek a piktogramov v programe. Pridáva to na konzistencii programu, používateľ nemá pocit že pracuje s dvojrozmerným elementom v trojrozmernom programe.

Aj keď Gravity Sketch ponúka odpovede na problémy, ktoré sa týkajú aj tejto práce, nie všetky riešenia sú priamo aplikovateľné. Je to z toho dôvodu že Gra-

vity Sketch je aplikácia naprogramovaná za použitia knižníc ponúkaných firmou Unity³, zatiaľ čo my používame webový rámec A-Frame. Vychádzajúc z porovnania WebVR a NativeVR v štúdiu od autorov M. Hemström a A. Forsberg, webová virtuálna realita stále neponúka toľko možností z hľadiska interakcie ako nativeVR.

1.4.2 Brushwork VR

Brushwork VR⁴ je aplikácia, ktorá ponúka možnosť simulácie maľovania na plátno vo VR. Narozdiel od Gravity Sketch, Brushwork VR je webová aplikácia, čím jej riešenia sa dajú považovať za relevantnejšie pre túto prácu z hľadiska možností, aké technológie pre vývoj webVR aplikácií dnes ponúkajú.

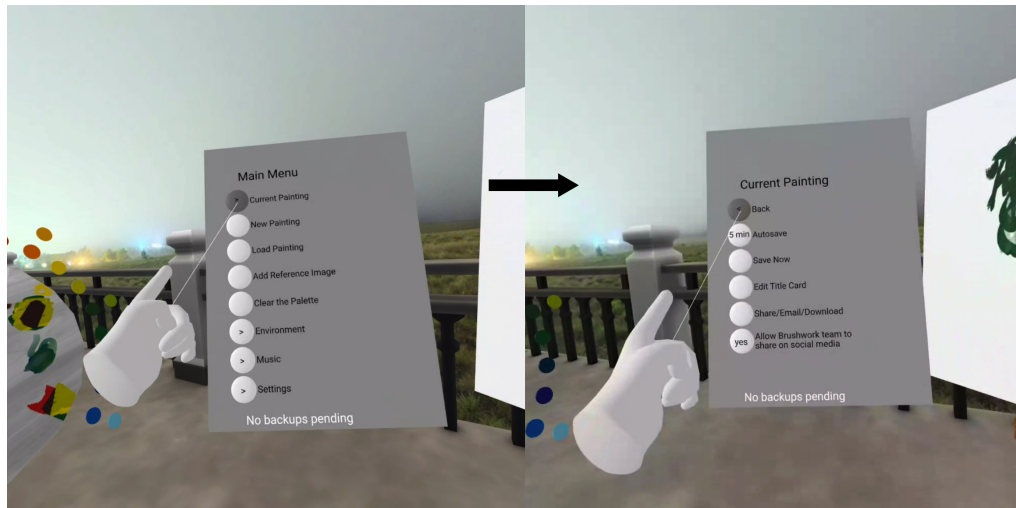
Brushwork VR má menu a nastavenia riešené formou statického panelu s tlačidlami^{1.12}. V prípade potreby má používateľ možnosť panel chytiť do ruky a presunúť ho na ľubovoľnú pozíciu. s panelom je možné pracovať aj pri jeho presúvaní. Elementy a interakcia sú riešené formou nápisov a tlačidiel. Ak si používateľ želá vnoriť do danej možnosti, stlačí tlačidlo vedľa názvu. Rovnako funguje interakcia na hlbších vrstvách v menu.

Nástroje, v tomto prípade štetce a paleta s farbami sú riešené formou reálnych objektov v ich životnej veľkosti, ktoré môže používateľ uchopiť do ľubovoľnej ruky a následne s nimi pracovať. Prepínanie hodnôt jednotlivých rozšírených možností štetcov sú mapované na tlačidlá na ovládačoch.

V porovnaní s inými VR aplikáciami, Brushwork VR má model interakcie s nástrojom navrhnutý trochu inak. V bežných VR programoch môžeme vidieť že ak chceme použiť nástroj, je potrebné ho chytiť do ruky a neustále držať, aj počas jeho používania. Keď používateľ tlačidlo držania pustí, nástroj spadne na zem. V Brushwork VR stačí stlačiť tlačidlo držania raz, pri chytení nástroja do ruky, a druhý krát pri jeho pustení. Toto riešenie je pre človeka intuitívnejšie a menej obmedzujúce. Medzi praktické črty programu patrí aj bezťažový stav. Používateľ si teda môže všetky predmety umiestniť na ľubovoľné miesto v priestore.

³<https://unity.com/unity/features/vr>

⁴www.brushworkvr.com



Obr. 1.12: Hlavné menu v aplikácii Brushwork VR

1.4.3 Zhodnotenie analýzy súčasného stavu VR aplikácií

Trh s VR aplikáciami je naozaj bohatý. Jeho diapazón siaha naprieč rôznymi žánrami, od pracovných nástrojov, cez kreatívne nástroje až po hry. Vo vybranej vzorke analyzovaných VR aplikácií sme sa primárne zameriavali na používateľský zážitok (UX), používateľské rozhranie (UI), intuitívnosť ovládania, riešenie problému interpretácie dvojrozmerných ovládacích prvkov v trojrozmernom priestore a na ponúknuté interakčné funkcionality a ich mapovanie. Zhrnutie kľúčových poznatkov z analýzy súčasného stavu VR aplikácií:

- **Interpretácia dvojrozmerných ovládacích prvkov prostredníctvom tabletu**
- toto riešenie vyššie predstaveného problému je praktické a ľahko rozšíriteľné, vďaka čomu ho môžeme vidieť v oboch analyzovaných aplikáciách.
- **Mapovanie funkcionalít na gestá v spojení s tlačidlami**

1.5 Analýza dostupných technológií

Pointou tejto časti analýzy je nájsť a otestovať technológie, o ktoré by bolo možné projekt NDMVR rozšíriť.

1.5.1 A-Frame htmlembded Component

Komponent `htmlembded`⁵ slúži na zobrazenie klasickej web stránky napísanej v HTML do A-Frame scény. V prípade tohto projektu je táto technológia potrebná na vy-

⁵<https://www.npmjs.com/package/aframe-htmlembded-component>

tvorenie dvojrozmerných ovládacích elementov za účelom interakcie so scénou. Je vytvorená ako komponent, zapadajúci do entity-component-system ⁶ vzoru navrhnutého vývojármi A-Frame. Pre jej použitie je potrebné pridať tento komponent do `<a-entity>`. Následne entita dokáže zobraziť v A-Frame scéne všetky html elementy, ktoré sú do nej vložené.

Pre otestovanie funkčnosti technológie sme nasadili do projektu jednoduchý experiment. Komponent má za úlohu zobraziť HTML tlačidlo v A-Frame scéne, ktoré vypíše počet kliknutí, ktoré boli naň uskutočnené. Testujeme teda nie len to, či technológia naozaj dokáže zobraziť HTML element v A-Frame scéne, ale aj možnosti interakcie s elementom, ktoré by podľa dokumentácie `htmlembed` komponentu mali byť prístupné. Implementácia je zobrazená v kóde 1.1.

Zdrojový kód 1.1: Implementácia komponentu `htmlembed`

```
...
return (
  <a-entity
    class='clickable'
    position='0.25 0 0'
    htmlembed
    onClick={rerender}
  >
    <TestButton />
  </a-entity>
)
```

Zdrojový kód 1.2: Tlačidlo pre zobrazenie v A-Frame scéne

```
function TestButton(){
  let count = 0;
  function handleClick() {
    count += 1;
  }
  return(
    <button onClick={handleClick}>
      Clicked {count} {count === 1 ? 'time' : 'times'}
    </button>
  )
}
```

⁶<https://aframe.io/docs/1.3.0/core/entity.html#sidebar>

Element sa nepodarilo do scény vložiť, a teda aj interakcia s ním bola nemožná. Program pri pokuse zobrazíť komponent zamrzol.

V druhom experimente sme sa snažili zobrazíť jednoduchý `<div>` element, bez potreby interakcie. Ako môžeme vidieť v kóde ??, elementu sme nastavili fixnú výšku a šírku, spolu s farbou pozadia, aby bol spozorovateľný.

Zdrojový kód 1.3: Testovací `<div>` element pre zobrazenie v A-Frame scéne pomocou `htmlembed` komponentu

```
function TestDiv(){
  return(
    <div style={{
      width: '100px',
      height: '100px',
      backgroundColor: '#FF0000'
    }}></div>
  )
}
```

Tento pokus sa tak isto nepodarilo uskutočniť. Program vykazoval rovnaké správanie ako v prípade prvého experimentu. Výpis z konzoly bol prázdny v oboch experimentoch.

Keďže uskutočnené pokusy o implementáciu `htmlembed` komponentu do projektu dopadli neúspešne, túto technológiu hodnotíme za nevyhovujúcu.

1.5.2 Zhodnotenie analýzy dostupných technológií

Naším primárnym záujmom v tejto časti analýzy bolo nájsť technológiu, ktorá by nám dovoľovala zobrazíť html kód na plošinu v A-Frame scéne s cieľom vytvorenia ľahko funkciono rozširiteľného tabletu s možnosťou interakcie s tlačidlami. Na internete sme našli niekoľko ponúknutých alternatívnych riešení, no všetky boli založené na komponente `htmlembed`. Preto sme sa rozhodli analyzovať len tento komponent. Jeho implementácia bola neúspešná, z čoho vyplýva že tento problém budem musieť riešiť iným spôsobom.

2 Syntetická časť

V tejto časti práce sa zameriame na návrh a konkrétnu implementáciu dostupných riešení. Keďže je tento projekt rozšírením projektu autora práce [2], technológie, ktoré budú použité v tomto projekte máme fixne dané. Riešenie bude implementované za použitia knižnice JSROOT¹ a web frameworks A-Frame² a ReactJS³.

2.1 Návrh implementácie Tabletů

Cieľom tejto bakalárskej práce je zjednodušenie interakcie s projektom NDMVR a zlepšenie jeho používateľského zážitku. Na základe analýzy existujúcich a dostupných riešení sme sa rozhodli implementovať do projektu takzvaný tablet, ktorý bude slúžiť ako interakčné médium používateľa s projektom. Tento komponent je principiálne inšpirovaný reálnym svetom. Jedná sa napríklad o scenár domu s inteligentným osvetlením, kedy používateľ má zariadenia, ktorých funkcionality nevie využiť priamo, fyzicky, ale musí na to použiť komunikačné médium vo forme buď priloženého ovládača, inteligentného tabletu, alebo aplikácie pre osobný smartphone. Komponent tablet v tomto projekte nie je samostatné inteligentné zariadenie, je to plocha, ktorá slúži na zobrazenie dvojrozmerných ovládacích prvkov projektu, ktoré sa v tomto momente buď nedajú implementovať do trojrozmernej (skratka 3D) scény vôbec, alebo ich implementácia by bola časovo a výkonnostne nerentabilná. Opierame sa hlavne o možnosti používaných technológií.

2.1.1 Princíp používania tabletu

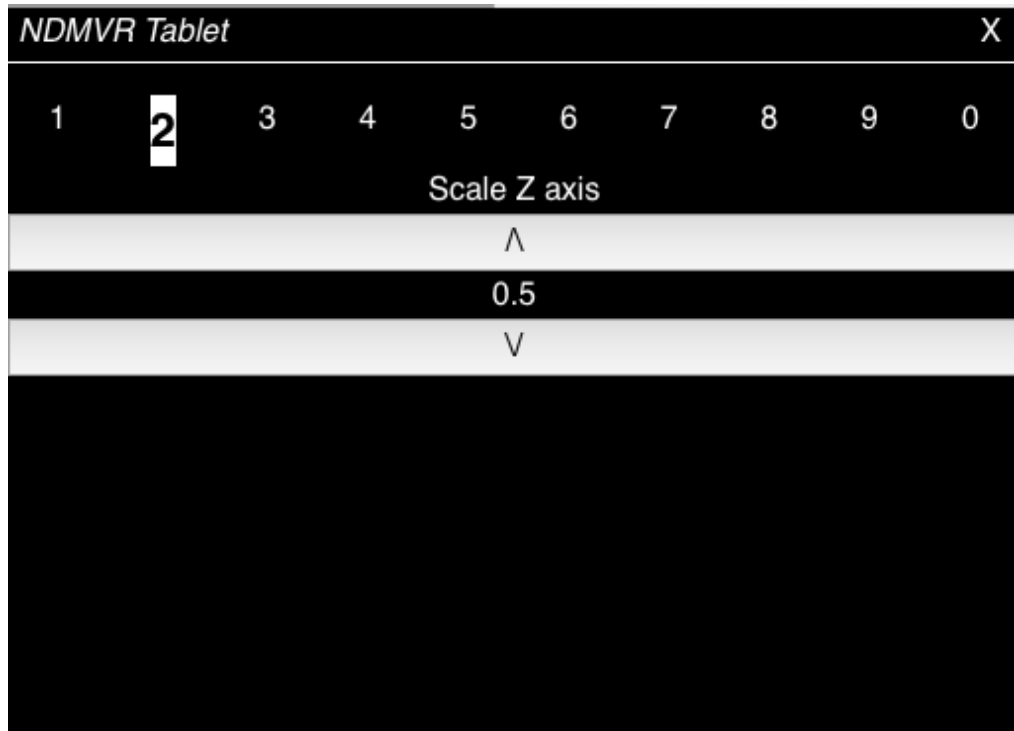
Na Obr. 2.2 môžeme vidieť diagram prípadov použitia tabletu. Tablet obsahuje lištu s desiatimi číslami usporiadanými tak, ako sú usporiadané na bežnej klávesnici v rade nad písmenami. Každé číslo reprezentuje unikátny nástroj, pomocou

¹<https://jsroot.gsi.de>

²<https://aframe.io>

³<https://reactjs.org>

ktorého vie používateľ vykonávať rôzne akcie s histogramom. Nástroje majú samozrejme aj svoje názvy, no v záujme efektívnej implementácie v rámci projektu má každý svoje unikátne číslo. Pod tým sa nachádza názov konkrétne zvoleného nástroja a jeho ovládacie prvky s informáciami. Používateľ si mení aktívny nástroj a pracuje s tabletom výhradne klikaním myšou.

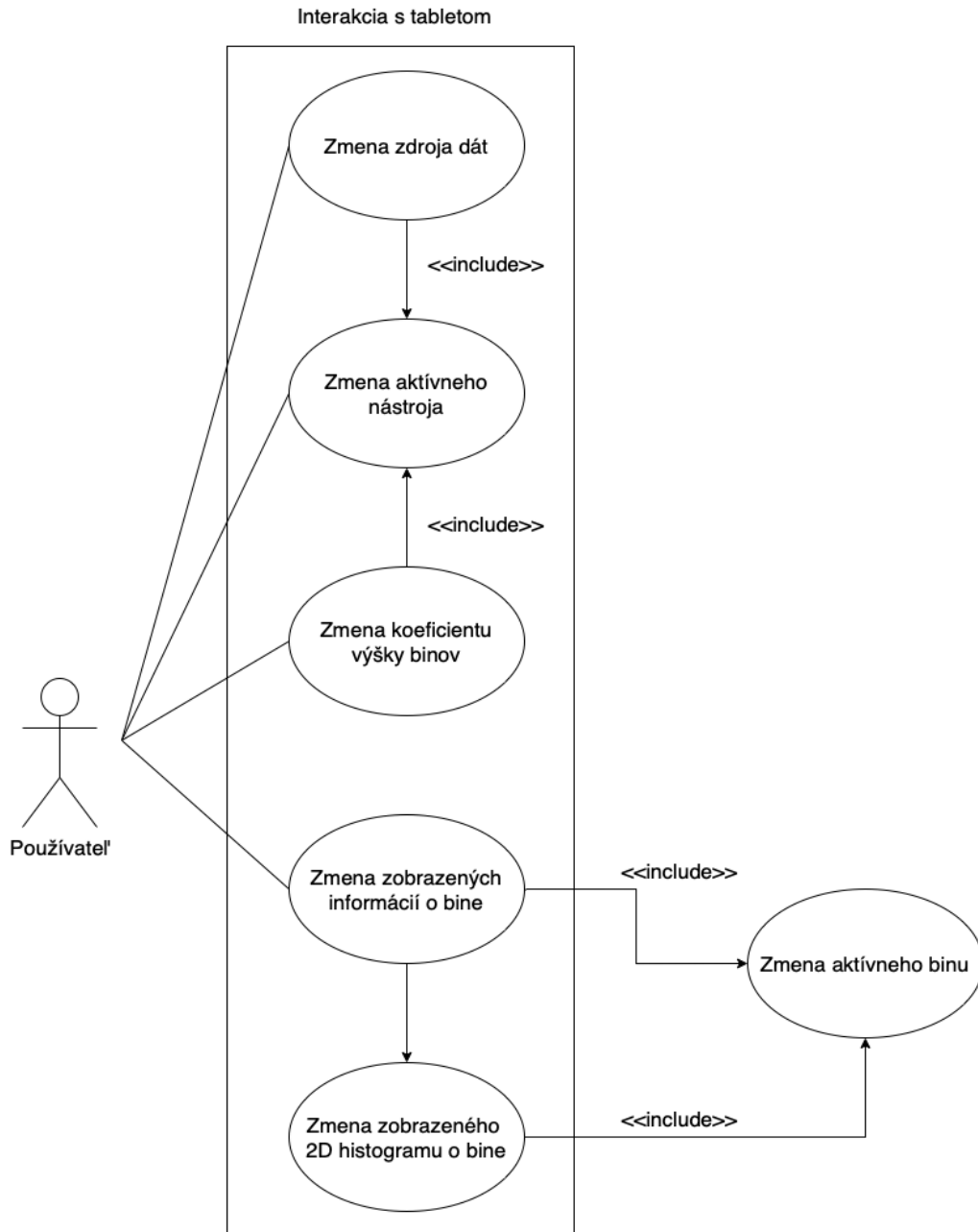


Obr. 2.1: Prvý prototyp tabletu pre desktopovú verziu webu. (Dizajn nie je finálny)

2.1.2 Problém s interakciou

Z dôvodu nízkej výpočtovej náročnosti a tak isto jednoduchej implementácie sme sa rozhodli implementovať tablet ako samostatný div kontajner nad A-Frame scénu. Po konzultácii s Martinom Fekete sme zistili, že toto riešenie je validné a plne vyhovujúce pre počítačového používateľa.

Problém nastal v prípade používateľov, ktorí by radi používali tento projekt prostredníctvom Oculus Quest 2 zariadenia. Plný potenciál zariadenia sa dá využiť len v prípade, že webová aplikácia podporuje VR mód. Ten síce podporuje A-Frame scéna, no pri jeho aktivácii scéna zakryje všetky ostatné elementy na stránke. Keďže sa náš tablet nachádza nad A-Frame scénou, po aktivácii VR módu sa tento tablet stáva nepoužiteľným.



Obr. 2.2: Diagram prípadov použitia tabletu.

2.1.3 Tablet pre VR zariadenia

Na to, aby bol tablet použiteľný vo VR móde, a teda na VR zariadeniach, je potrebné ho implementovať do A-Frame scény. Na základe analýzy existujúcich riešení sme sa rozhodli tento tablet pripevniť k ruke, alebo teda k ovládaču používateľa vo virtuálnej realite. Idea spočíva v ovládaní tabletu jednou rukou, pomocou tlačidiel na ovládači, zatiaľ čo druhá ruka slúži na interakciu so scénou. Podľa [13] 90% populácie sú praváci, preto sme sa rozhodli pripevniť tablet na sekundárnu ruku väčšiny obyvateľstva, aby pravá ruka mohla slúžiť na motoricky jemnejšiu

prácu v scéne, ako je napríklad výber binov.

Keďže väčšina tlačidiel na oboch ovládačoch má už určenú vlastnú funkcionality, bolo potrebné navrhnuť kombinácie tlačidiel pre ovládanie tabletu. Aby bolo ovládanie čo najintuitívnejšie, rozhodli sme sa sprístupniť ovládanie tabletu so stlačením tlačidla pod prostredníkom. Toto rozhodnutie je odvodené z reálneho sveta, kde keď chce človek pracovať s tabletom, musí ho uchopiť do ruky a neustále držať, inak vplyvom gravitácie tablet padne na zem. Tento ovládací prvok by mal byť pre používateľa jednoduchý na pochopenie a ľahký na zapamätanie. Keď je toto tlačidlo stlačené, zvyšok rozhrania ovládača je plne k dispozícii na interakciu s tabletom. Konkrétna funkcia jednotlivých interakčných elementov na ovládači priamo závisí od práve aktívneho nástroja. Výnimkou sú 3 tlačidlá:

- tlačidlo pod prostredníkom
- tlačidlo Y
- tlačidlo X

Funkcionalita týchto tlačidiel sa v závislosti od práve aktívneho nástroja nemení. Ako je už vyššie spomenuté, tlačidlo pod prostredníkom slúži ako spúšťač interakcie s tabletom. Tlačidlo Y slúži na zvolenie nasledujúceho nástroja z ponuky za aktívny, a tlačidlo X slúži na zvolenie predchádzajúceho nástroja. Po dosiahnutí hraničného nástroja sa aktivuje prvý nástroj na opačnom konci výberu. Teda ak je aktívny nástroj číslo 1 a používateľ stlačí tlačidlo X, aktívnym nástrojom sa stane ten s číslom 10, a naopak.

2.1.4 Funkcionalita a nástroje dostupné prostredníctvom tabletu

Ako je vyššie spomenuté, tablet obsahuje rad nástrojov a funkcionalít, ktorými umožňuje používateľovi interakciu so zobrazenými dátami. Tablet zatiaľ obsahuje tieto nástroje:

- Zobrazenie informácií o bine
- Menenie mierky pre os Z
- Zobrazenie 2D reprezentácie aktuálne zvoleného binu
- Volenie zdroja dát

Zobrazenie informácií o bine - Tablet zobrazí používateľovi rozšírené informácie o práve označenom bine. V prípade kliknutia na iný bin, informácie na tablete sa obnovia.

Menenie mierky pre os Z - Pomocou implementovaných tlačidiel si používateľ vie zmeniť koeficient výšky zobrazených binov. Hodnota je prednastavená na 0.5, a so spomínanými tlačidlami si vie používateľ zmeniť túto hodnotu s krokom 0.2.

Zobrazenie 2D reprezentácie aktuálne zvoleného binu - Na tablete sa zobrazí 2D histogram aktuálne označeného binu. Je to ten istý histogram, ako sa zatiaľ zobrazuje aj na plátne za 3D histogramom.

Volenie zdroja dát - Tablet ponúkne používateľovi menu s množinou dostupných zdrojov dát na zobrazenie v 3D histograme.

2.1.5 Návrh implementácie nástrojov

V záujme čo najjednoduchšej rozšíriteľnosti ponuky dostupných nástrojov sme navrhli implementáciu funkciami. Každý nástroj bude naprogramovaný ako funkcia, ktorá pri zavolaní vykoná svoju úlohu a následne vráti celý obsah obrazovky tabletu pre vykreslenie. Konkrétna forma obrazovky pre vykreslenie sa bude líšiť na základe toho, či používateľ pristupuje k funkcii prostredníctvom zariadenia Oculus Quest, alebo prostredníctvom klasického počítača. Okrem naprogramovania nástroja bude potrebné ho len pridať do jedného zoznamu s nástrojmi, a bude pripravený na použitie. Navrhnutá implementácia je zobrazená v pseudo kóde 2.1.

Zdrojový kód 2.1: Pseudo kód implementácie nástroja.

```
//pole obsahuje odkazy na nástroje
const tools = [getBinInfoTool(), bin2Dchart(), Nastroj(),
               scaleZaxis(), sourceSelect()]

function Nastroj(){

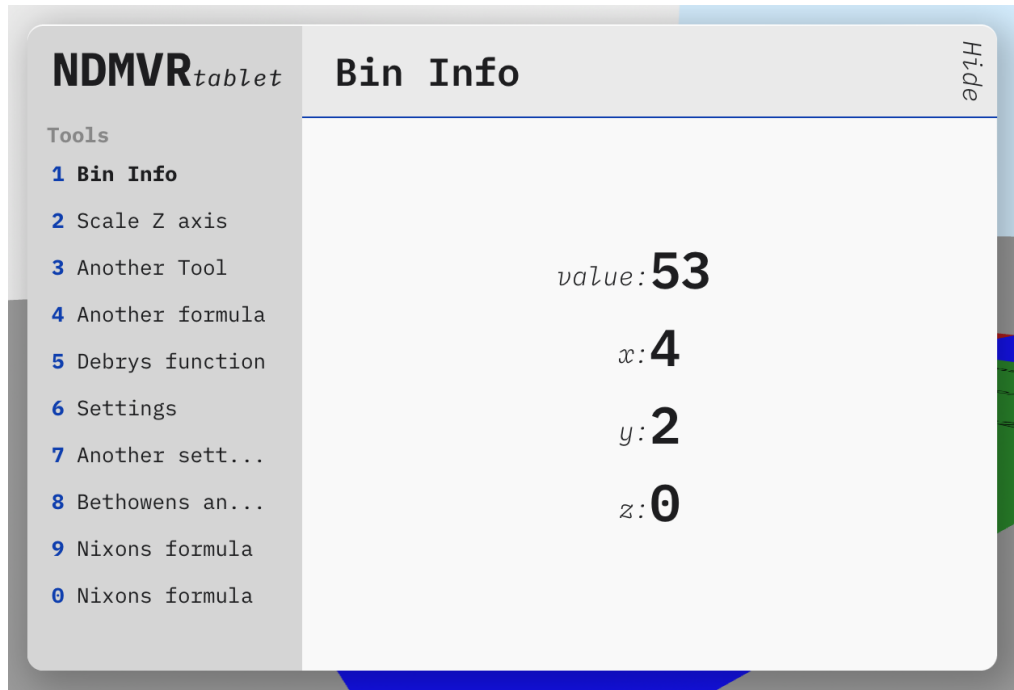
    //priestor pre telo kodu

    if(AFRAME.utils.device.checkHeadsetConnected() === "desktop"){
        return(
            //komponenty pre zobrazenie vystupu z nástroja na tablete
            //pre desktop verziu aplikacie
        )
    }else{
        return(
            //komponenty pre zobrazenie vystupu z nástroja na tablete
```

```

        //pre Oculus verziu aplikacie
    )
}
}

```



Obr. 2.3: Dizajn NDMVR tabletu.

2.1.6 Vizuálny návrh tabletu

Tablet je dvojrozmerné používateľské rozhranie, ktoré ako element umožňuje používateľovi ho zavrieť, otvoriť, zmeniť veľkosť alebo presunúť na ľubovoľnú pozíciu v rámci obrazovky. Týmto interakčnými možnosťami je veľmi podobný bežnému oknu v operačnom systéme. Práve preto sme sa pre dizajn tabletu 2.3 inšpirovali bežným oknom v operačnom systéme. Toto rozhodnutie vie v praxi priniesť niekoľko výhod.

Podľa slov Christiana Crumlish a Erina Malone[14] používatelia preferujú dizajny, v ktorých registrujú poznané vzory. V tomto prípade je práve podobnosť s oknom v operačnom systéme rozoznateľným vzorom. Prvou výhodou je pozitívnejší používateľský zážitok na základe rozoznateľného vzoru.

Systémové okná v prípade väčšiny operačných systémov podporujú okrem iných možností presúvanie okna, minimalizovanie okna, zmenu dimenzií okna. Cieľom nášho dizajnu je na základe rozpoznanie tohto vzoru očakávať interakčné možnosti prislúchajúce systémovému oknu. Očakávaným efektom je vedomie

používateľa o týchto možnostiach aj napriek tomu že dizajn o ich existencii priamo neinformuje.

Ako nosné farby dizajnu sme zvolili modrú, sivú, bielu a čiernu. Modrá v tomto dizajne slúži ako akcent. Podľa [15] modrá je farbou technológií. Hlavnou úlohou akcentovej farby v dizajne je vyzývať používateľa k akcii. Farbu sme sa rozhodli aplikovať práve na očíslovanie nástrojov z toho dôvodu, že každé číslo označuje zároveň aj klávesovú skratku ktorou si vie používateľ daný nástroj aktivovať. Očíslovanie nástrojov je generované automaticky, na základe poradia nástrojov v poli `tools[]` v komponente `DesktopTablet`. V prípade ak by si používateľ chcel zmeniť poradie nástrojov, je potrebné upraviť ich pozíciu v danom poli.

Sivá farba sa viaže na sekundárne prvky alebo doplnujúce informácie, ako je sekcia so zoznamom nástrojov.

Biela farba svojim vysokým kontrastom na seba prirodzene púta pozornosť, preto je aplikovaná ako podklad pre telo aktívneho nástroja.

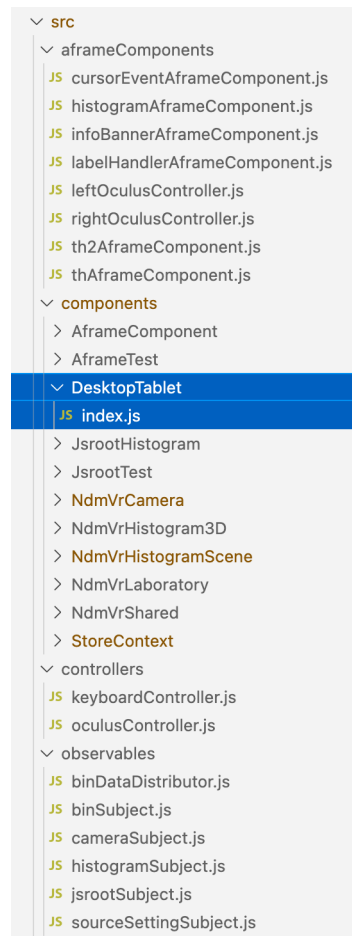
Čierna farba je použitá ako farba nosných textov, z dôvodu vysokého kontrastu a teda dobrej čitateľnosti so všetkými podkladovými farbami nástroja.

2.2 Implementácia tabletu

Táto práca nadväzuje po softvérovej stránke na prácu [2]. Z toho dôvodu bolo nevyhnutné dodržať technológie, v ktorých je komponent realizovaný a tak isto potrebné je dodržať zaužívané konvencie v projekte. Na obr. 2.4 je zobrazený výsek obsahu adresára `src` z projektu `NDMVR`, kde môžeme vidieť spôsob implementovania komponentov, ktorí bol zvolený autorom pri vytváraní projektu. Každý komponent má svoj adresár a v ňom súbor `index.js`, ktorý obsahuje samotný kód komponentu. Pre dodržanie konvencie sme implementovali tablet rovnakým spôsobom (je zvýraznený modrou farbou v obr. 2.4). Tablet komponent je v adresárovej štruktúre nazvaný `DesktopTablet` a v záujme zachovania izolovanosti komponentov a modularity kódu sa plná implementácia tabletu nachádza v práve jednom súbore. Závislosti `DesktopTablet-u` je možné vidieť v komponentovom diagrame v obr. 2.6.

2.2.1 Komponent `DesktopTablet()`

Tablet implementujeme ako samostatný komponent. Nástroje sú implementované formou funkcií. Každý nástroj, na to aby bol plne funkčný musí mať vlastnú funkciu, byť zapísaný v zozname nástrojov a jeho návratová hodnota musí obsahovať 2 objekty; Jeden objekt je používateľské rozhranie nástroja v prípade, že po-

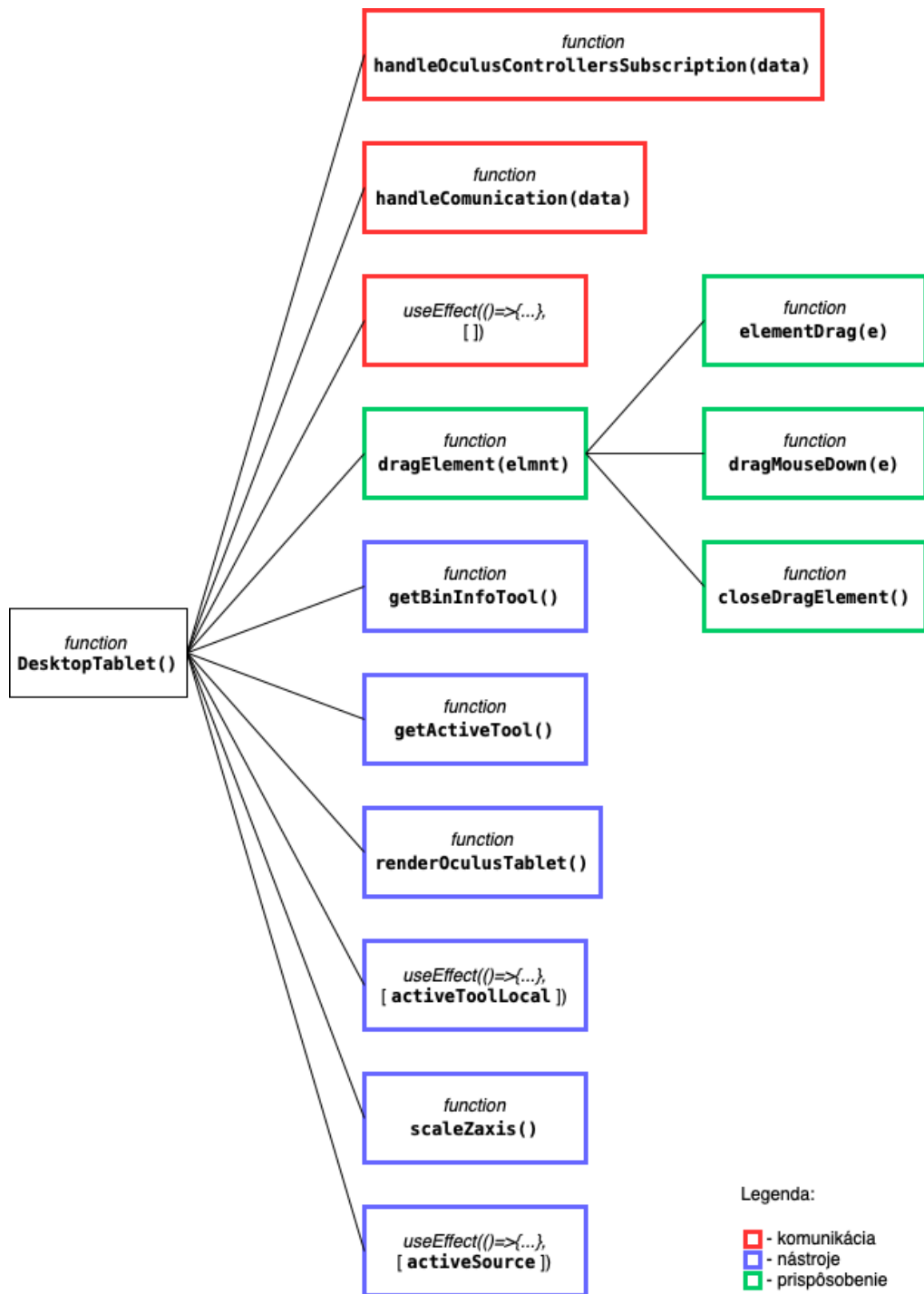


Obr. 2.4: Stromová štruktúra adresára src projektu NDMVR.

užívateľ prístupuje k programu prostredníctvom desktopu a druhý pre prípad prístupu za použitia zariadenia Oculus Quest 2.

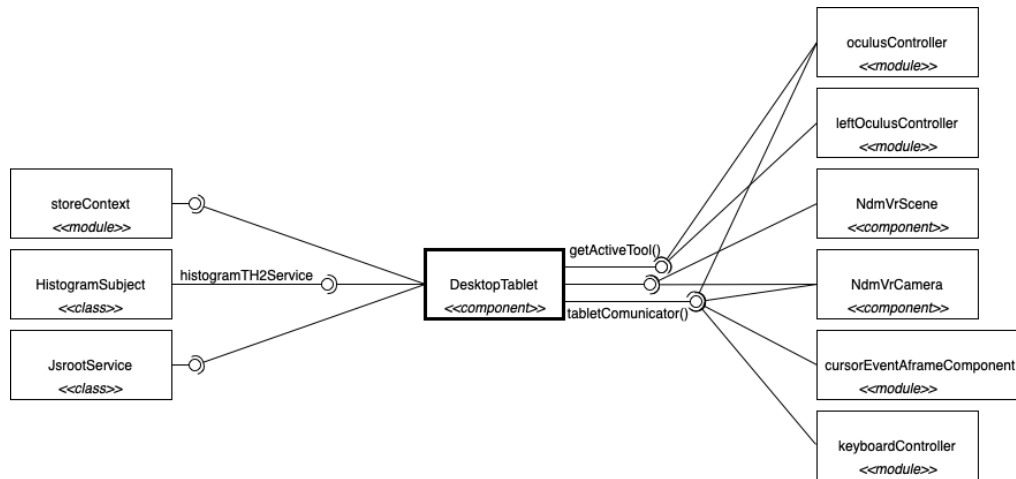
Tablet neprijíma žiadne vstupné parametre a vracia svoje používateľské rozhranie buď ako `<div>` element pre desktopovú verziu, alebo `<React.Fragment>` pre Oculus.

Na základe pokusu o prenesenie tabletu, navrhnutého pre desktopovú verziu aplikácie, s jeho plnou funkcionalitou a interakčnými možnosťami do A-Frame scény sme zistili, že táto implementácia je veľmi náročná a hlavne spôsobuje zamrznutie programu. Konkrétne v prípade nástroja číslo 2, slúžiaceho na zmenu koeficientu pre zobrazenú výšku binov. Po vykonaní danej operácie v niektorých prípadoch stránka zamrzla na dobu až 20 sekúnd. V niektorých prípadoch program spadol úplne. Na základe analýzy chybového hlásenia sme zistili, že chyba sa nenachádza v kóde, ktorý implementuje danú funkcionalitu, ale v internom kóde knižnice A-Frame. Postupným odstraňovaním jednotlivých častí nástroja vysvitlo že problém spôsoboval výpis aktuálnej hodnoty koeficientu na tablet vo VR scéne. Na základe tohto faktu sme dospeli k rozhodnutiu odstrániť zo-



Obr. 2.5: Obsah komponentu DesktopTablet.

brazenie hodnoty aktuálneho koeficientu výšky binov z nástroja a minimalizovať vizuálne a interakčné možnosti tabletu v A-Frame scéne na minimum. Realizáciou týchto krokov sa nám podarilo úspešne znížiť výpočtovú náročnosť programu. Boli sme nútení upustiť od pôvodného plánu sprístupniť novú funkcionality pre používateľa v A-Frame scéne s čo najlepším používateľským zážitkom, a zamerali sme sa len na sprostredkovanie nových možností bez ohľadu na fi-



Obr. 2.6: Komponent diagram DesktopTabletu a relevantných komponentov.

nálny používateľský zážitok.

React Hook-u `useState()` funguje ako premenná, ktorej zmena vie vyvolať re-render elementu v ktorom je použitá. Komponent `DesktopTablet()` používa následovné inštancie `useState()`:

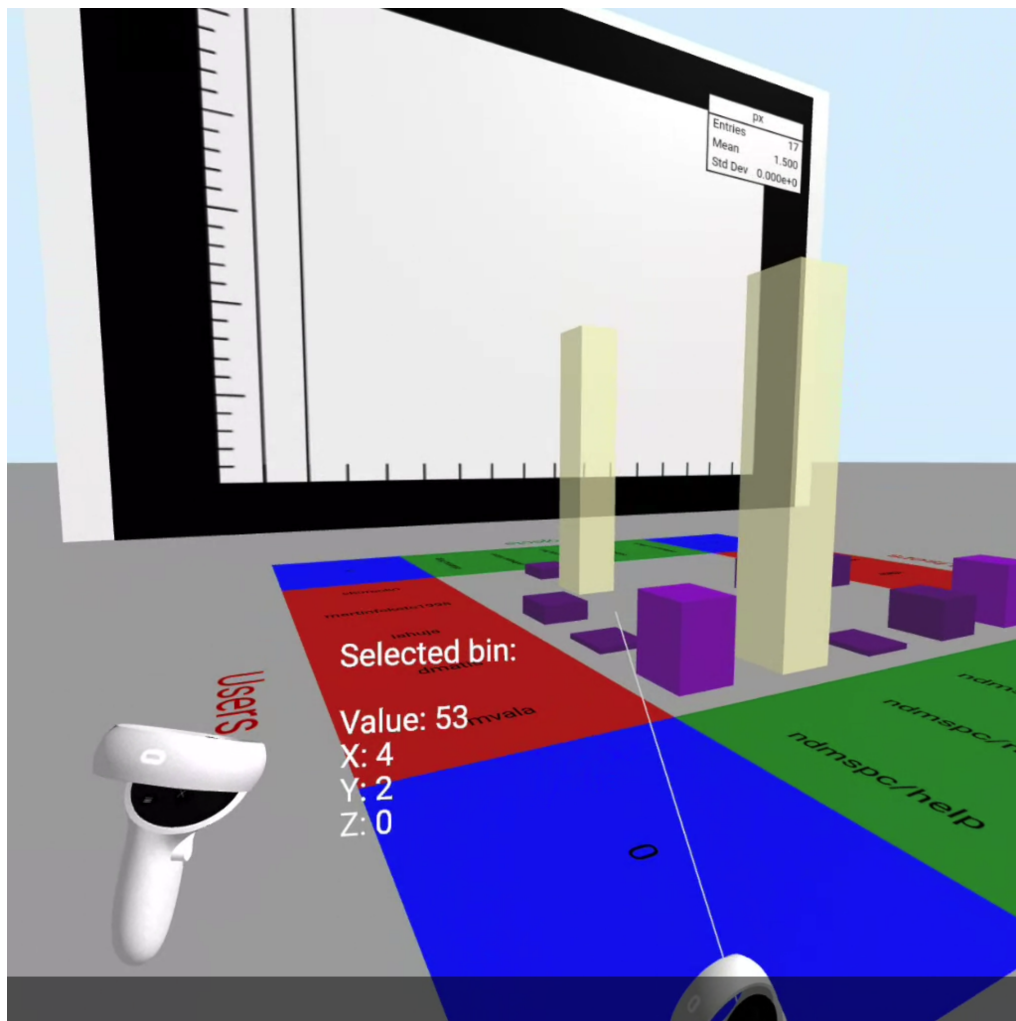
- **activeToolLocal** - obsahuje číslo práve aktívneho nástroja
- **activeSource** - nesie systémový názov práve aktívneho zdroja dát
- **binInfo** - pozostáva z informácií o označenom bine. Konkrétne jeho pozíciu v priestore a jeho hodnotu. Dáta má uložené vo forme `key:value` objektu
- **scaleCoeficient** - obsahuje z hodnoty koeficientu, ktorý je súčasťou vzorca pre výpočet grafickej výšky binov. `tablet`
- **tabletDisplay** - obsahuje aktuálnu hodnotu `css` vlastnosti `display`, ktorá je použitá na zobrazenie alebo schovanie tabletu pre desktop verziu nástroja.
- **activeDevice** - Pozostáva z informácie o vstupnom zariadení. Môže nadobudnúť len hodnotu `'oculus'` alebo `'desktop'`. Jej hodnota sa nastaví pri spustení programu a počas chodu programu sa nemení.

Tablet sa skladá z nasledovných funkcií (vizuálnu reprezentáciu aj so všeobecným kategorizovaním funkcií opisuje obr. 2.5):

- **handleCommunication(data)** - spracováva prichádzajúcu komunikáciu z externých komponentov. Parameter `data` je `key:value` a obsahuje informáciu poslanú externým komponentom. Nemá žiadnu návratovú hodnotu.

- **handleOculusControllersSubscription(data)** - funkcia nastavuje novú hodnotu koeficientu výšky binov pre nástroj `scaleZaxis()`. Parameter `data` obsahuje požadovaný prírastok k aktuálnej hodnote koeficientu. Funkcia nemá žiadnu návratovú hodnotu.
- **useEffect(()=>...,[])** - tento `useEffect` hook sa spustí pri počiatočnom vykreslení komponentu. Jeho úlohou je inicializovať odber dát z komunikačného kanálu a presmerovať tok dát do funkcie `handleCommunication`. Návratovou hodnotou je funkcia, ktorá odhlasuje komponent z odberu informácií z komunikačného kanálu.
- **renderOculusTablet()** - funkcia vracia celé používateľské rozhranie aktívneho nástroja vo forme A-Frame komponentov na základe hodnoty objektu `activeToolLocal`. Na vstupe neprijíma žiadnu hodnotu a v prípade, že nedisponuje nástrojom pre aktuálnu hodnotu `activeToolLocal` vráti chybovú hlášku tak isto vo forme A-Frame komponentu.
- **getBinInfoTool()** - funkcia vráti `<div>` element so zobrazenými informáciami o bine z objektu `binInfo`. Na vstupe neprijíma žiadne argumenty.
- **scaleZaxis()** - nastavuje novú výšku binov v 3D histograme podľa vstupu od používateľa. Vracia interface pre zmenu koeficientu výšky zobrazených binov vo formáte `<div>` elementu. Na vstupe neprijíma žiadnu hodnotu.
- **getActiveTool()** - návratovou hodnotou je číslo aktívneho nástroja. Neprijíma na vstupe žiadnu hodnotu.
- **useEffect(()=>..., [activeToolLocal])** - funkcia nastaví elementu s `id='tablet'` obsah na obrázok s 2D reprezentáciou aktuálne označeného binu. Reaguje na každú zmenu objektu `activeToolLocal` ale vykoná sa len v prípade, ak je aktívny nástroj č.1 a zároveň používateľ prístupuje k programu prostredníctvom zariadenia Oculus. Neprijíma ani nevracia žiadnu hodnotu.
- **useEffect(()=>..., [activeSource])** - zmení zdroj dát pre hlavný histogram v A-Frame scéne na aktuálnu hodnotu v objekte `activeSource`. Reaguje na zmenu hodnoty spomínaného objektu. Nevracia ani neprijíma žiadnu hodnotu.
- **dragElement(elmnt)** - sprístupňuje možnosť presunúť `elmnt` na ľubovoľné miesto na obrazovke. Prijíma parameter `elmnt`, ktorý je referenciou na objekt v Dokument Objekt Modeli stránky.

- **dragMouseDown(e)** - získava aktuálnu polohu kurzora na obrazovke po stlačení ľavého tlačidla myši. Vstupný parameter e je udalosť, ktorá vznikne pri stlačení tlačidla myši.
- **elementDrag(e)** - udržiava presúvaný element pod myšou počas toho ako je myš stlačená a používateľ s ňou hýbe. Vstupný parameter e je udalosť, ktorá vznikne pri stlačení ľavého tlačidla myši. Nevracia žiadnu hodnotu.
- **closeDragElement()** - ukončuje pohyb elementu po uvoľnení tlačidla myši. Nevracia ani neprijíma žiadnu hodnotu.



Obr. 2.7: Zjednodušená verzia desktopového tabletu pre VR zariadenia.

2.2.2 Komunikácia tabletu s inými komponentami

Pre maximalizovanie jednoduchosti a efektivity kódu používame na asynchrónnu komunikáciu tabletu s inými komponentami projektu práve jednu inštanciu ob-

jektu `Subject()` z knižnice `rxjs`⁴. Inštanciu sme nazvali `tabletComunicator`. Prostredníctvom tohto komunikátora môžeme v ľubovoľnom komponente projektu poslať dáta tabletu. Aby tablet po prijatí informácie vedel o aké dáta sa jedná, je potrebné ich pred poslaním obaliť do `key:value` objektu. V tablete sa následne dosadia jednotlivé hodnoty.

2.2.3 Zefektívňovanie komunikácie s tabletom

Pri testovaní daného riešenia pre komunikáciu komponentov s tabletom sme zistili, že pri stlačení ľubovoľného tlačidla na Oculus ovládačoch príslušný controller komponent vyšle práve dva signály tabletu namiesto jedného. Ani po komplexnejšom debugovaní sa nám nepodarilo prísť na koreň problému, preto sme tento problém vyriešili pomocou premennej `bugFixer` v `DesktopTablet()` komponente, pomocou ktorej tablet spracováva len každý druhý signál ktorý dostane a každý prvý zahodí. Implementácia je zobrazená v pseudo kóde 2.2.

Zdrojový kód 2.2: `bugFixer` aplikovaný v kóde

```
useEffect(() => {
  bugFixer = !bugFixer
  if ((bugFixer && activeDevice === 'oculus') ||
    activeDevice === 'desktop') {
    tabletComunicator.subscribe(handleCommunication)
  }
  return () => tabletComunicator.unsubscribe()
}, [])
```

2.2.4 Zmena v implementácii nástrojov

Po implementácii tohto návrhu však vo verzii projektu pre VR zariadenia snímková frekvencia klesla na neakceptovateľne nízke hodnoty. Spomalenie je dôsledkom toho, že v tomto návrhu sa pri každej interakcii s tabletom prekresľuje celý tablet, a nie len zmenené dáta. Hľadiac na túto skutočnosť sme boli nútení zmeniť model implementácie nástrojov do tabletu.

Nový návrh obsahuje funkciu `renderOculusTablet()`, ktorá vykreslí používateľské prostredie v tablete pre aktuálne zvolený nástroj pri zvolení nástroja a nástroj následne dosadzuje a mení len hodnoty. Tento model je síce zložitejší na pochopenie pre používateľa, ktorý si chce vytvárať nové nástroje, ale úspešne zrýchlil odozvu programu pri interakcii s tabletom.

⁴<https://rxjs.dev>

2.2.5 Funkcia `renderOculusTablet()`

Táto funkcia zabezpečuje vrátenie rozhrania aktívneho nástroja pre tablet v A-Frame scéne. Obsahuje príkaz `switch()`, ktorý reaguje na hodnotu uloženú v `activeToolLocal`. Funkcia vracia celé rozhranie aktívneho nástroja a jej renderovanie spúšťa zmena hodnoty `activeToolLocal`. V prípade, ak má `activeToolLocal` nastavenú takú hodnotu, ktorá nemá svoj pridelený nástroj, funkcia vypíše v tablete chybovú hlášku. Chod programu sa nenaruší.

2.2.6 Implementácia tabletu pre desktop

Implementácia návrhu tabletu pre desktopovú verziu nemala žiaden dopad na plynulosť programu. V záujme rozšírenia interakčných možností pre používateľa sme pridali možnosť zvolenia nástroja prostredníctvom klávesových skratiek. Na zvolenie konkrétneho nástroja okrem priameho kliknutia na jeho číslo sa dá použiť aj dané číslo na klávesnici. Druhou akciou s dvoma spôsobmi vykonania je otvorenie a zatvorenie tabletu. To sa dá realizovať buď prostredníctvom tlačidla "Hide" v hornom rohu tabletu, alebo za použitia tlačidla E na klávesnici. Rozhodli sme sa túto funkcionality pripísať na tlačidlo E z dôvodu, že sa nachádza hneď vedľa písmena W, ktoré v rámci kríža tlačidiel W,S,A,D slúži na pohyb v scéne a teda je umiestnené na ľahko dostupnom mieste.

Zdrojový kód 2.3: `tools[]`.

```
const tools = [
  { fun: getBinInfoTool(), name: 'Bin Info' },
  { fun: scaleZaxis(), name: 'Scale Z axis' },
  { fun: funkcia(), name: 'Menovka nástroja' }
]
```

Desktop verzia tabletu vracia svoj štandardný interface ako `<div>` objekt a na základe čísla aktívneho nástroja volá príležitú funkciu z poľa `tools[]` (obr. 2.3) ktorá poskytuje meno nástroja pre zobrazenie používateľovi a tak isto vracia interface samotného nástroja.

2.3 Pridávanie nástroja

Zdrojový kód 2.4: Pseudo kód implementácie nástroja.

```
const renderOculusTablet = () => {
```

```

switch (activeToolLocal) {
  case 0:
    return (
      //komponenty pre zobrazenie uzivatelskeho prostredia
      //pre tablet v Oculus verzii aplikacie
    )

//pole obsahujuce odkazy na nastroje
const tools = [getBinInfoTool(), bin2Dchart(), Nastroj(),
               scaleZaxis(), sourceSelect()]

function Nastroj(){

  //priestor pre telo kodu

  if(AFRAME.utils.device.checkHeadsetConnected() === "desktop"){
    return(
      //komponenty pre zobrazenie vystupu z nastroja na tablete
      //pre desktop verziu aplikacie
    )
  }else{
    return(
      //vystupne hodnoty nastroja
      //pre Oculus verziu aplikacie
    )
  }
}
}

```

2.3.1 Pridávanie nového nástroja

Pre zadefinovanie nového nástroja a jeho prístupnenie v oboch verziách tabletu sú potrebné nasledujúce kroky:

- Pridanie príkazu case do funkcie switch() vo funkcii RenderOculusTablet() s návratovou hodnotou obsahujúcou interface pre Oculus verziu tabletu
- Vytvorenie funkcie, ktorá vráti interface pre desktopovú verziu tabletu
- vloženie referencie funkcie spomenutej v predchádzajúcom kroku do poľa tools[] v tvare fun:nazovFunkcie(), name:"Názov funkcie pre zobrazenie používateľovi

Tento postup je zobrazený tak isto na obrázku 2.4.

3 Vyhodnotenie

V tejto kapitole práce sa zameriame na používateľské testovanie implementovaného riešenia a jeho vyhodnotenie a celkové vyhodnotenie riešenia.

3.1 Používateľské testovanie

Po dosiahnutí celistvého riešenia prichádza na rad používateľské testovanie. Cieľom riešenia je rozšíriť možnosti interakcie s nástrojom, urobiť tieto možnosti čo najintuitívnejšie na použitie a to so zreteľom aj na používateľský zážitok. Oslovili sme vzorku respondentov, ktorí mali za úlohu vypracovať krátky scenár v programe a na základe ich nadobudnutých skúseností a pocitu z programu vyplniť dotazník. Scenár pozostával z nasledujúcich úloh:

- Otvorte tablet a presuňte ho na ľubovoľnú pozíciu na obrazovke
- Zmeňte farebnú schému na nature
- Zmeňte zdroj dát pre zobrazený histogram na cern
- Zmeňte veľkosť zobrazeného výseku na 10 binov
- Zmeňte škálu zobrazených binov na osi Z na 0.1
- Prejdite sa okolo alebo skrz zobrazené dáta
- Zobrazte si ľubovoľný bin v kine a na bočnom paneli
- Pozrite si jeho hodnotu v tablete

V dotazníku sme sa rozhodli použiť dve metriky. Prvou metriku je SUS (System Usability Scale - z angličtiny metrika použiteľnosti systému)¹, pretože je konštruovaná na hodnotenie použiteľnosti programu. Druhou je NASA TLX (Task Load in-

¹<https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

deX - index zaťaženia úlohou).², ktorej zameranie je smerované na meranie rôznych druhov záťaže, ktorým bol respondent pri naplňaní scenáru vystavený.

Používateľského testovania sa zúčastnilo 11 respondentov, z toho väčšina boli študenti informatiky na vysokej škole.

3.1.1 Vyhodnotenie SUS metriky

Projekt NDMVR dosiahol v používateľskom testovaní skóre 72,5. V rámci SUS testovania, skóre vyššie ako 68 sa berie ako nadštandardné, čo znamená že projekt NDMVR uspel v teste použiteľnosti a teda účastníci testovania boli s nástrojom prevažne spokojní. Aj keď skóre 72,5 je celkom dobré, stále je tu priestor na zlepšenie.

3.1.2 Vyhodnotenie NASA TLX metriky

Po spriemerovaní výsledkov projekt NDMVR získal hodnotenie 46,49. Toto číslo popisuje na škále od 0 do 100 všeobecnú záťaž vytvorenú projektom. To znamená že čím nižšie číslo, tým lepšie. Z tohto skóre vieme usúdiť že v projekte je čo zlepšovať. Každopádne vieme dedukovať že výsledky sú jemne skreslené faktom že väčšina účastníkov nemala žiadne skúsenosti s dátovou analýzou a teda neznalosť domény zvýšila záťaž na vypracovanie scenára.

3.1.3 Vyhodnotenie používateľského testovania

Metriky nám ukázali že projekt disponuje priestorom na zlepšenie. Dôležitým faktorom presnosti výsledkov používateľského testovania je aj počet zúčastnených. Keďže sa testovania zúčastnilo len 11 respondentov, môžeme výsledky považovať za nie úplne precízne.

²<https://humansystems.arc.nasa.gov/groups/tlx/>

4 Záver

Cieľom práce bolo rozšíriť interakčné možnosti pre prácu s histogramom v projekte NDMVR s dôrazom na intuitívnosť ovládania a používateľský zážitok. Téma používateľského rozhrania vo webVR nie je zatiaľ dostatočne prebádaná a preto má táto práca skôr experimentálny charakter. Tak isto technológie, ktoré bolo potrebné zlúčiť a simultánne využívať pre vývoj neboli určené pre takéto spojenie čo neraz vyvršilo do neočakávaných komplikácií a obmedzení, čo nám nedovolilo si presne definovať ciele, ktoré v práci chceme dosiahnuť.

V práci sme prvotne na základe analýzy nástroja NDMVR zistili nedostatky používateľského rozhrania, následne sme v analýze existujúcich riešení zistili ako nativeVR aplikácie riešia túto problematiku a ako webVR aplikácie riešia túto problematiku. Spoznali sme používané technológie a otestovali perspektívne technológie na vyriešenie problému. Následne bol vyhotovený návrh riešenia, ktorý sa opiera o poznatky nadobudnuté z analýzy. Počas implementácie sme narazili na niekoľko obmedzení, na základe ktorých bolo nevyhnutné riešenie okresať a pozmeniť. Nakoniec sa nám aj napriek komplikáciám podarilo realizovať celistvé riešenie, ktoré v používateľskom testovaní získalo jemne nadpriemerné skóre.

Práca obohatila projekt NDMVR o nový komponent, tablet, ktorého pointou je poskytnúť intuitívne dvojrozmerné používateľské rozhranie pre ovládacie prvky programu a pre nástroje na prácu s dátami. Tablet obsahuje tiež sadu nástrojov, ktoré sprístupňujú používateľovi radu nových možností medzi ktoré patria napríklad zmena výšky zobrazených binov v histograme, možnosť zmeny zdroja dát pre histogram vo virtuálnej realite, zobrazenie 2D reprezentácie aktuálne označeného binu priamo na ovládači vo virtuálnej realite, alebo zobrazenie rozšírených informácií na tablete o práve označenom bine. K implementácii nástrojov sme zaujali skôr kvalitatívny ako kvantitatívny postoj. Tablet okrem týchto nástrojov priniesol do projektu úplne novú platformu pre používateľa na zadefinovanie vlastných nástrojov s unikátnym spôsobom interakcie ako v prípade jej desktopovej verzie, tak aj vo verzii pre zariadenia Oculus Quest 2.

Literatúra

1. PEARSON, Karl. *Philosophical Transactions of the Royal Society of London. A, Volume 186, pp. 343-414.* 1895. ISBN 1895RSPTA.186..343P.
2. FEKETE, Martin. *Vizualizácia experimentálnych údajov v zdieľanej rozšírenej realite a jej používateľské rozhranie.* 2021. Technická univerzita v Košiciach Fakulta elektrotechniky a informatiky.
3. FARSHID, Mana; PASCHEN, Jeannette; ERIKSSON, Theresa; KIETZMANN, Jan. Go boldly!: Explore augmented reality (AR), virtual reality (VR), and mixed reality (MR) for business. *Business Horizons.* 2018, roč. 61, č. 5, s. 657–663.
4. JSROOT. [B.r.]. Dostupné tiež z: <https://jsroot.gsi.de>.
5. VANDERDONCKT, Jean; PRIBEANU, Costin. State of the art of web usability guidelines. 2005.
6. CARMIGNIANI, Julie; FURHT, Borko; ANISETTI, Marco; CERAVOLO, Paolo; DAMIANI, Ernesto; IVKOVIC, Misa. Augmented reality technologies, systems and applications. *Multimedia tools and applications.* 2011, roč. 51, č. 1, s. 341–377.
7. *Introduciton into A-FRAME.* [B.r.]. Dostupné tiež z: <https://aframe.io/docs/1.2.0/introduction/>.
8. THREEJS. [B.r.]. Dostupné tiež z: <https://threejs.org>.
9. ZEISS VR ONE Plus - Virtual Reality Brille For Smartphone. [B.r.]. Dostupné tiež z: <https://www.amazon.de/ZEISS-VR-ONE-Plus-Smartphone/dp/B01GHN88HU>.
10. *Take a Virtual Reality tour of six REAL exoplanets.* [B.r.]. Dostupné tiež z: <https://www.youtube.com/watch?v=qhLExhpXX0E>.
11. *Project NEO: Virtual Reality Data Visualization for Machine Learning.* [B.r.]. Dostupné tiež z: <https://www.youtube.com/watch?v=myI4P9C34A0>.

12. *Gravity Sketch now free for private users.* [B.r.]. Dostupné tiež z: <https://www.vrnerds.de/gravity-sketch-jetzt-kostenlos-fuer-private-anwender/>.
13. MASUD, Yusra; AJMAL, M Asir et al. Left-handed people in a right-handed world: A phenomenological study. *Pakistan Journal of Social and Clinical Psychology*. 2012, roč. 10, č. 1, s. 49–60.
14. CRUMLISH, Christian; MALONE, Erin. *Designing social interfaces: Principles, patterns, and practices for improving the user experience.* " O'Reilly Media, Inc.", 2009.
15. WHITFIELD, TW; WHILTSHIRE, TJ. Color psychology: a critical review. *Genetic, social, and general psychology monographs*. 1990.

Zoznam skratiek

2D Dvojmerný.

3D Trojmerný.

AR Rozšírená realita.

MR Zmiešaná realita.

UI Používateľské prostredie.

UX Používateľský zážitok.

VR Virtuálna realita.

Zoznam príloh

Príloha A Používateľská príručka

Príloha B CD médium – záverečná práca v elektronickej podobe.

Obsah prílohy B:

- Adresár src - obsahuje projekt NDMVR so všetkými zdrojovými súbormi
- Adresár doc - obsahuje túto prácu, používateľskú príručku a anonymizované výsledky používateľského testovania vo formáte PDF
- Adresár tex - pozostáva zo zdrojových kódov dokumentov v adresári doc

Technická Univerzita Košice
Fakulta elektrotechniky a informatiky

Možnosti vizualizácie experimentálnych údajov vo webovej rozšírenej realite

Používateľská príručka

Inštalácia a spúšťanie a vypnutie projektu NDMVR

Ešte pred inštaláciou samotného projektu je potrebné si nainštalovať nasledovné nástroje:

- Npm
- Node.js

Následne môžeme pokračovať k inštalácii samotného nástroja:

1. Skopírujeme celý obsah adresára src z prílohy B do počítača
2. Otvoríme nový terminál v hlavnom adresári projektu
3. Zadáme príkaz **npm run reset**
4. Zadáme príkaz **npm run start**
5. Otvoríme si nové terminálové okno a premiestnime sa do adresára example
6. Zadáme príkaz **npm run start**

V prípade ak je proces úspešný, v terminálovom okne adresára example sa zobrazí takýto výpis:

```
Compiled successfully!
```

```
You can now view ndmvr-example in the browser.
```

```
Local:           https://localhost:3000  
On Your Network: https://192.168.0.127:3000
```

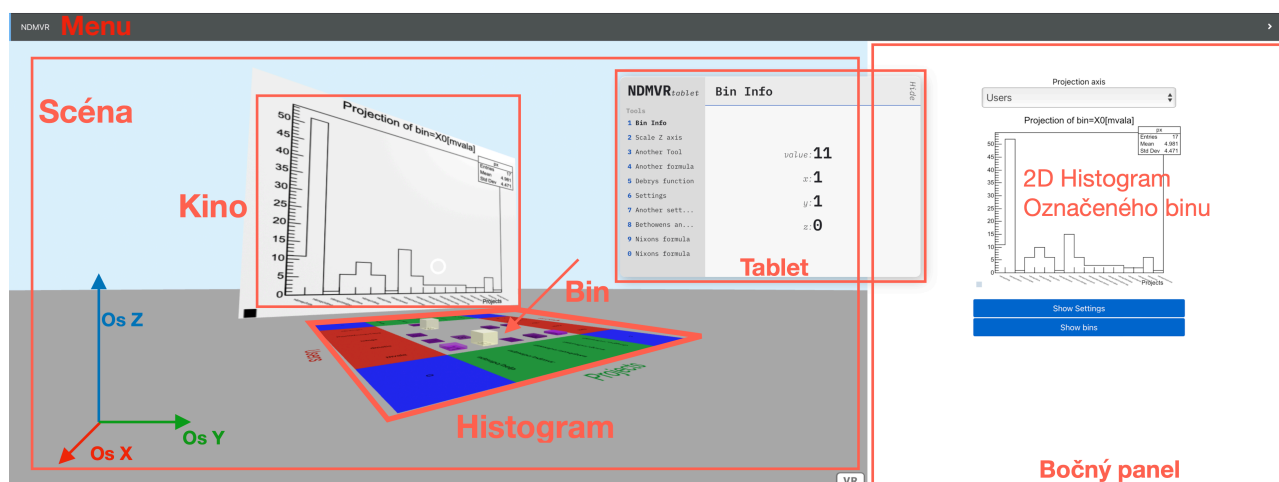
```
Note that the development build is not optimized.  
To create a production build, use npm run build.
```

```
□
```

V tomto momente sa nám zobrazí nástroj otvorený vo webovom prehliadači. V tomto bode vieme k nástroju prísť aj použitím zariadenia Oculus Quest 2. Je ale potrebné aby bolo pripojené v tej istej sieti ako počítač. Pre spustenie nástroja v zariadení Oculus Quest 2 stačí použiť link z terminálu ktorý sa nachádza za správou “On Your Network:”.

Prvé 3 kroky postupu sú inštalačné. Od 4. kroku je popísaný proces spúšťania. Pre vypnutie programu stačí stlačiť klávesovú skratku Ctrl + C v oboch oknách terminálu.

Vysvetlenie projektu NDMVR



Po rozkliknutí menu môžeme vidieť tieto možnosti:

- TH3Demo
- TH3Example
- TH2Example

Podstránka TH2Example má k dispozícii naširší výber zdrojov dát. Z toho dôvodu pri skúšaní nástroja odporúčame použiť TH2Example.

Ovládanie Desktop verzie nástroja NDMVR

Tablet komponent sa otvára a zatvára použitím tlačidla 'E'. Je možné ho premiestniť na ľubovoľnú pozíciu na obrazovke a pomocou pravého dolného rohu sa dajú zmeniť jeho dimenzie.

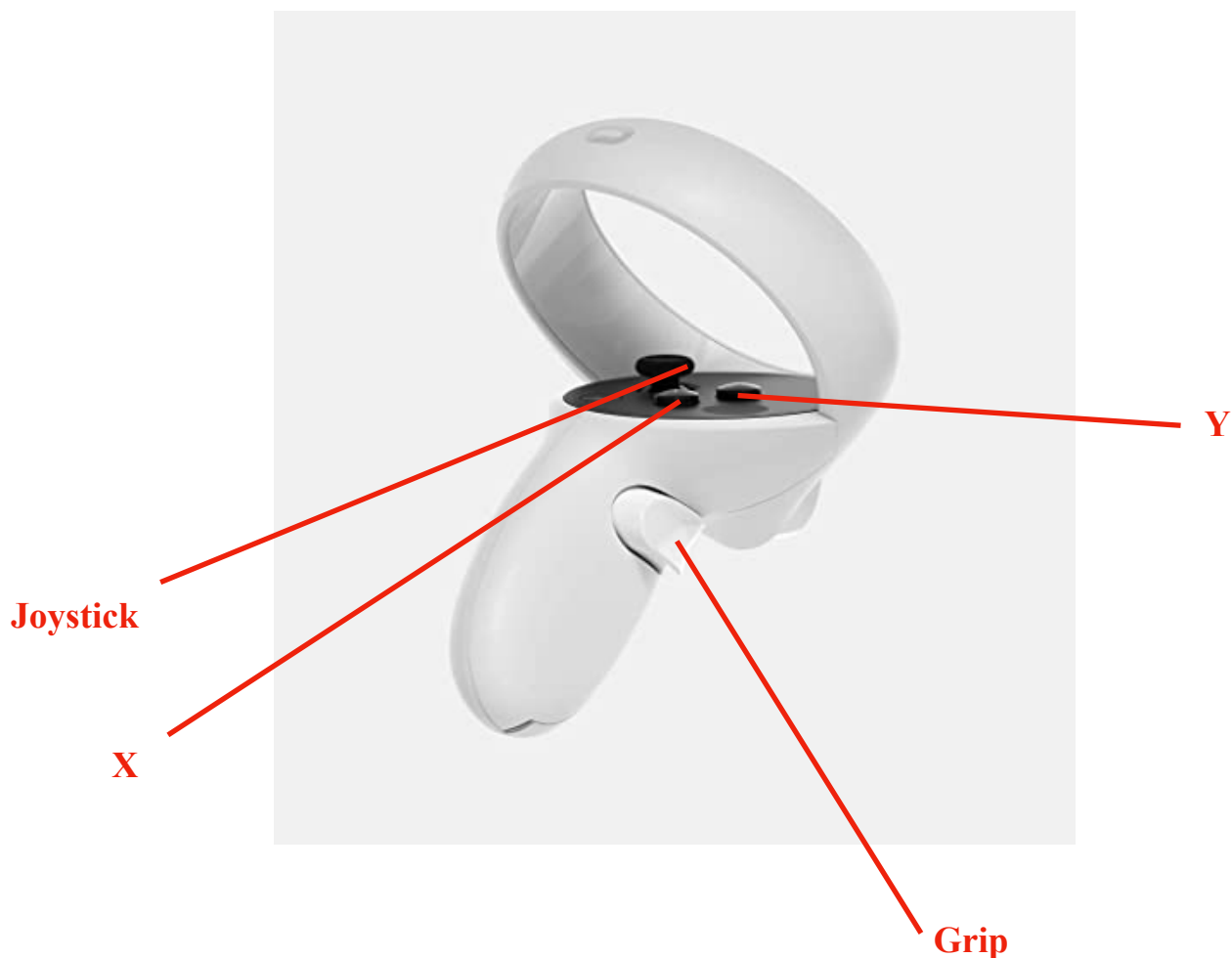
Klávesové skratky a možnosti sa zobrazia po stlačení klávesy 'V'.

Pre označenie binu naň stačí kliknúť. Následne sa jeho 2D reprezentácia zobrazí na komponente kino a tak isto aj v bočnom paneli.

(poznámka: priamo po spustení nástroja sa plná funkcionálna sprístupní až po kliknutí na ľubovoľné 2 biny.)

Tlačidlo VR spustí scénu v imerzívnom móde. Desktop verzia nástroja nie je konštruovaná pre tento mód.

Ovládanie Oculus verzie nástroja NDMVR



Verzia nástroja NDMVR pre Oculus Quest 2 je konštruovaná len pre imerzívny mód.

Pre veškerú interakciu s tabletom je nevyhnutné držať tlačidlo Grip.

Ovládanie tabletu v Oculus verzii NDMVR

Grip + Y = nasledujúci nástroj. Ak sa nachádzame na poslednom nástroji, stlačenie tejto kombinácie zobrazí prvý nástroj

Grip + X = predchádzajúci nástroj. Ak sa nachádzame na prvom nástroji, stlačenie tejto kombinácie zobrazí posledný nástroj

Grip + Joystick Dopredu / Grip + Joystick Dozadu = Týmito kombináciami ovládame nástroj na zmenu výšky binov a nástroj pre zmenu zdroja dát.

Mapovanie ostatných funkcií a možností sa zobrazia po stlačení tlačidla 'Y'.

Vývoj Oculus verzie aplikácie NDMVR

Pre zobrazenie vývojárskej konzoly z prehliadača v zariadení Oculus je potrebné vykonať nasledujúce kroky:

1. Nainštalovať Google Chrome do počítača (iný prehliadač nie je podporovaný)
2. Spustiť projekt NDMVR
3. Prepojiť káblom zariadenie Oculus Quest 2 v počítačom
4. V zariadení Oculus sa zobrazí okno s otázkou či chceme povoliť počítaču prístup k dátam. Stlačíme tlačidlo Allow.
5. Spustíme v zariadení Oculus projekt NDMVR.
6. V prehliadači Google Chrome navštívime túto adresu: <chrome://inspect/#devices>
7. Zo zoznamu kariet otvorených na zariadení Oculus si vyberieme tú s názvom NDMVR a stlačíme tlačidlo 'inspect'.

Remote Target #LOCALHOST

Quest 2 #1WMHH840A70482

com.oculus.browser (OculusBrowser/21.0.0.2.38.366647150 SamsungBrowser/4.0 Chrome/100.0.4896.127) [trace](#)

Panel App Nav UI <chrome://panel-app-nav/>
[inspect](#)

Oculus browser chrome://oculus-ntp/?oculus_browser_addressbar_input_type=1&oculus_browser_animate_addressbar=1&ocul...
[inspect](#) [pause](#)

Panel App Nav UI <chrome://panel-app-nav/>
[inspect](#)

Panel App Nav UI <chrome://panel-app-nav/>
[inspect](#)

Oculus browser chrome://oculus-ntp/?oculus_browser_addressbar_input_type=1&oculus_browser_animate_addressbar=1&ocul...
[inspect](#) [pause](#)

ndmvr <https://ndmspc.gitlab.io/ndmvr/#/aframe/thProjection>
[inspect](#) [pause](#)

Následne sa nám otvorí okno s vývojárskou konzolou.